Jussi Piitulainen

Explorations in the distributional and semantic similarity of words

Academic dissertation to be publicly discussed,
by due permission of the Faculty of Arts at the University of Helsinki
in auditorium XIV, on the 15th of January, 2011 at 10 o'clock.

ISBN 978-952-92-8426-9 (paperback)
ISBN 978-952-10-6760-0 (PDF)

http://ethesis.helsinki.fi

Helsinki 2010
Helsinki University Print

Explorations in the distributional and semantic similarity of words

Jussi Piitulainen

Abstract

A straightforward computation of the list of the words (the 'tail words' of the list) that are *distributionally* most similar to a given word (the 'head word' of the list) leads to the question:

> How *semantically* similar to the head word are the tail words; that is: how similar are their *meanings* to its meaning? And can we do better?

The experiment was done on nearly 18 000 most frequent nouns in a Finnish newsgroup corpus. These nouns are considered to be distributionally similar to the extent that they occur in the same direct dependency relations with the same nouns, adjectives and verbs. The extent of the similarity of their computational representations is quantified with the information radius.

The semantic classification of head–tail pairs is intuitive; some tail words seem to be semantically similar to the head word, some do not. Each such pair is also associated with a number of further distributional variables. Individually, their overlap for the semantic classes is large, but the trained classification-tree models have some success in using combinations to predict the semantic class.

The training data consists of a random sample of 400 head–tail pairs with the tail word ranked among the 20 distributionally most similar to the head word, excluding names. The models are then tested on a random sample of another 100 such pairs. The best success rates range from 70% to 92% of the test pairs, where a success means that the model predicted my intuitive semantic class of the pair. This seems somewhat promising when distributional similarity is used to capture semantically similar words.

This analysis also includes a general discussion of several different similarity formulas, arranged in three groups: those that apply to sets with graded membership, those that apply to the members of a vector space, and those that apply to probability mass functions.

Contents

List of Figures

List of Tables

Preface

This book is a case study in the distributional similarity of words. The data set consists of nearly twenty thousand nouns that occur often in a Finnish newspaper corpus. The question is how their computed distributional similarity relates to their loose similarity of meaning. The answer is to look inside the distributional similarity judgments, and to model intuitive semantic judgments in terms of co-occurrence statistics.

The corpus and the parser Two resources became available in our department just when I needed them. The corpus was obtained from the publisher, and Mickel Grönroos worked on putting it in a standard format. Pasi Tapanainen and Timo Järvinen wrote their dependency parser for Finnish, inspired by their experience with Fred Karlsson's shallower grammar formalism, called constraint grammar. I made use of both of these resources when they were still new.

My tools I wrote my own suite of programs for the calculation of distributional similarity lists. These programs have evolved through the years, but the main method of calculation has not changed since I first had to cope with data sets that did not fit in the working memory of the computers at the time: the lists have a maximum length after which the least similar words simply drop out. Such lists can be computed on the subsets of the vocabulary; the results are then merged, with truncation to the maximum length.

The similarity lists that we study in this book took approximately two weeks to compute on two servers. Running the programs required a semi-complicated shell script to coordinate. No second run has been attempted, but both the computers and the programs have improved since.

Other uses of distributional similarity Distributional similarity is interesting in its own right, notably in the smoothing of the co-occurrence frequencies, see (Dagan et al., 1995). One overview of different applications

is by Julie Weeds (2003), pages 22–36. The present study is concerned only with distributional similarity as a substitute for semantic similarity.

Notes on the use of a parser I depended heavily on a morpho-syntactic parser, which added a significant amount of linguistic information to the corpus. This might be seen as distorting the authentic language data and thus making the results suspect. Here I adopt a different view that the parser merely makes explicit such relations as might already be apparent to a human observer.

Note that some level of analysis is necessary to identify the tokens that are referred to as words, and to deal with whatever markup there may be in addition to the actual text. Hindle (1990) went further:

> The stumbling block to any automatic use of distributional patterns has been that no sufficiently robust syntactic analyzer has been available.

Even I experimented once with this same data without using the dependency links. Instead, co-occurrence was defined as an occurrence within a short window of tokens. The resulting similarity lists looked distinctly less convincing, but this has to remain a supporting anecdote only.

The use of parsers appears to be a common practice in the field. Gregory Grefenstette used dependency relations in his thesis (Grefenstette, 1994). Dekang Lin (1998b) likewise incorporated them in his work and he notes three others in the beginning of his Section 5. Lillian Lee also used a parser (1999) when comparing similarity formulas.

Acknowledgments

Initially, far too many years ago, I was thinking of doing something completely different. One day, Kimmo Koskenniemi, professor of computational linguistics and my advisor, showed me Grefenstette's book (1994) and suggested that someone should do something along those lines here, too. I still do not quite know if he was thinking that I might be that somenone, but here I am, still trying to understand what this is all about. I have slowly learned all that I know about statistical methods while working on this.

Essential advice Koskenniemi made another suggestion that I later took: to try the recursive partitioning methods. Their use required me to classify my distributional word pairs semantically. I tried to do this intuitively but with great care, and I could not. Many pairs were simply impossible for me

to decide in such a way that I would not be tempted to reverse the decision, and then re-reverse it again. In short, I was stuck.

Eventually, I confessed that I was stuck on just this point. Instead of dismissing my intention, Koskenniemi suggested a three-way classification: the pairs that I accept, the pairs that I reject, and those that were difficult to decide. Antti Arppe, who was then finishing his own thesis, provided another piece of advice. He advised me to classify the pairs *fast* and to never go back on my decisions.

I was then able to finish my training set of 400 pairs. When, in the end, I added a test set of 100 more pairs, I knew what to do.

Miikka Silfverberg, who was hacking finite-state transducers next door, gave me crucial advice about Lebesgue integrals so that I could understand some of Sibson (1969). Specifically, he managed to convey to me that 'simple function' is a technical term. That helped me unravel the relevant parts of Rudin later.

A helpful question Late last century, I gave a talk in Arbeitsbereich NatS in Fachbereich Informatik of University of Hamburg. I showed my similarity tables for `omena`, *apple*, and `appelsiini`, *orange*, among other data. Someone in the audience asked why `vero#uudistus`, *tax reform*, appeared there. I have no idea who you are, but thank you for that question, as I looked at the underlying data afterwards. Now I know the answer, and I found it instructive, so I made it a running example in this book.

Prior usage Krister Lindén has already made use of these similarity lists in his thesis (Lindén, 2005; Lindén and Piitulainen, 2004). In the present book, I will examine their formation, and then re-assign them a new purpose.

Juha Makkonen and I wrote a paper (Makkonen and Piitulainen, 2001) where we used my similarity programs to expand terms. The programs were still awkward to use at the time. Later, as a small part of a larger project, Sirke Viitanen experimented with my programs, and that was what finally motivated me to improve my user interface. My thanks to both of you.

Chapter 1

Introduction

Take any two words and observe their uses in some large body of text. The words could be `apple` and `orange`, for example, and the text could come from the internet:

```
a magic APPLE was said to keep people young forever
planting APPLE trees that provided food and a livelihood
an overview of the ORANGE tree, and how to care for your own
Harvest ORANGES when they taste sweet.
```

(These do not constitute a representative sample, but they are actual usage on the internet.)

The two words are said to have a similar distribution, or to be distributionally similar, insofar as they occur in similar contexts. It is not necessary to restrict *context* to refer to the surrounding *text*, but even when we do, it is necessary to note that the *whole* context is almost never the same. We can still observe similar elements in the contexts where the two words occur, such as the word `tree` that occurs both with `apple` and with `orange` in the above examples.

If we first find a way to quantify just how distributionally similar any given word is to `apple`, or any other word, we can then find the words that are *most similar* to it, in some given body of text. All we need to do is to rank all words by the number that indicates the degree of similarity with the word `apple`. This can be done in practice, with some reservations, though there are many details that need to be considered.

Distributional similarity is of interest partly as a feasible substitute for semantic similarity, which is a similarity of meaning. I chose `apple` and `orange` as examples partly because the two words are, to some extent, semantically similar: both name kinds of fruit. The strongest form of semantic similarity

is likely to be synonymy, the relation between words that mean more or less the same. Much looser similarities will be recognised in this book.

The other reason for my choosing `apple` and `orange` was to protest the strange prohibition that one should never compare apples and oranges. On a more serious note, I did *not* choose them because they are either a particularly good example or a particularly bad example of distributional or semantic similarity. The example data in this book are either more or less arbitrary or are randomly sampled in the hope that the data would be representative of some larger population under investigation.

Distributional similarity is not the same as semantic similarity unless the latter is *defined* as the former, which it is *not*. Instead, I establish certain specific, if simple, approximations of the two notions, and then I observe to what extent they agree. First, I carry out this plan in the usual way. Next, I set up a new experiment of my own and see if I can separate the semantic wheat from the chaff among the distributionally best ranked pairs, using a trained procedure to match my intuitive semantic classification of a random sample of pairs.

The input to the semantic classifier consisted only of the different combinations of observed counts. These were available in the distributional ranking method adopted here, but they must have been underused because they appear to improve the ranking lists afterwards, semantically speaking. Further work is needed to determine how significant this improvement is in practice. It is also not known to what extent it still happens if one applies certain simple heuristic precautions early in the process, such as the removal of words that seem to be numerical outliers.

This study analyses Finnish data, so the actual example words are not `apple` and `orange`, but their Finnish counterparts `omena`, *apple*, and `appelsiini`, *orange*. Further, a morpho-syntactic parser is used to identify the words, the words that occur in direct dependency relations with them, and the base forms of the words. For example:

> `kuten OMENAT` ***`kuoritaan`*** `säilöntäaineiden takia`
> `ja` ***`kuorimme`*** `APPELSIINEJA Algarvessa`

The parser identifies `omenat` as the nominative plural of the noun `omena`, and `appelsiineja` as the partitive plural of the noun `appelsiini`. The parser also identifies `kuoritaan` as the present tense passive and `kuorimme` as a first person plural of the verb `kuoria`, *peel*. Finally, the parser identifies the noun as the direct object of the verb in both fragments, giving the two dependency triples:

```
kuoria-obj-omena
kuoria-obj-appelsiini
```

(Notation: `head-relation-dependent`) From both of these triples, I extract `kuoria-obj-` as an element that occurred together with the noun, which I call a (computational) *attribute* of the noun. The inclusion of the dependency relation is essentially just a technical detail, though the use of a parser is worth investigation. The attributes could have been limited to other words, such as the word *tree* in our initial examples.

Again, since the two nouns share an attribute, they are to some degree distributionally similar. To quantify the degree of similarity, a large corpus is analysed to count all such co-occurrences and to apply a certain mathematical formula to all such counts for the two words. To find the most similar words for every word, the words are all ranked by the number that indicates their degree of similarity. In this way, each word becomes the head word of a similarity ranking list (henceforth referred to as a 'similarity list' for short) of other words, in the order of decreasing similarity. To see how often such distributionally most similar words are semantically similar to the head word of the list, a random sample was analysed. Finally, to better identify the semantically similar words among the lists of distributionally most similar ones, classification procedures are trained and their success rates observed, which seem somewhat promising.

1.1 Research problems

My research problems, as they evolved during the work, can be summarised as the following four.

1. My objective was to understand the distributional similarity of words as they occur naturally in text, and the relationship to their semantic similarity; I approach this topic from a computational point of view.

2. I took a particular interest in the various similarity formulas I encountered in the literature. As a consequence, a chapter is devoted to a conceptual overview of these from a unified point of view. The focus is on 'bare' formulas, mostly ignoring the complications of the different weighting schemes, the methods where the main idea is some transformation of the whole space of word representations, and various clustering methods. Special attention is paid to the information radius formula used in the empirical part of this work.

3. I computed a relatively large-scale similarity table of frequent nouns in a Finnish newspaper corpus, based on syntactically determined co-occurrences and simple frequency weights. One focus is on the review of the actual co-occurrence patterns that made nouns appear similar.

4. I developed a new method for the further processing of similarity ranking lists of words, so that semantically good and bad entries could be identified. This exercise builds on the similarity data of the previous point. The new method involves a trained classifier and requires a small number of semantic input judgments in the training stage. After the training stage, the classifier uses the distributional information already available in the word representations in the distributional similarity calculations. The information radius is one of the input variables, the relative ranks come from the similarity table, and several more that can be computed from the pair of word representations.

The starting point Words that occur in the same contexts are said to have similar distributions, or to be *distributionally similar*. Words with close meanings are said to be *semantically similar*. The two notions appear to overlap so that distributionally similar words are also often semantically similar.

When 'occurrence in a context' is simplified to 'occurrence with other words', the result is a practical notion of word similarity that can be *computed* from the corpus data. Furthermore, similarities can be quantified and words ranked according to their similarity to a word. This gives us the means to extract, from text, the pairs of words that are distributionally similar in a specific sense, and so they may be semantically similar.

This only works to a certain extent. There will be pairs that turn out to be noise: either there are distributionally similar pairs that turn out to be not semantically similar after all, or the computable notion I adopt of distributional similarity is overly simplified.

The scope of this work The present work consists of applying the usual methods to a large number of frequent nouns in a Finnish newspaper corpus, with a novel attempt to train a classification tree to identify the semantically promising or suspicious pairs of words among the distributionally most similar pairs in the resulting ranking lists. The input to the classification tree comes from the same distributional data which were already available on a large scale when the ranking lists were created. The output is a semantic label, `good` or `bad`, for each pair. I added these labels manually to a set of training pairs and later to a smaller set of testing pairs.

The goal is to understand better what the corpus-based computational distributional similarity of words is. In preparation for the experiments, I will review both the main background concepts (in this chapter) and some known similarity formulas (Chapter 2). In the first stage of the experiments, I will cover in some detail both the corpus and the parser, then analysein

some detail the characteristics of the resulting distributional word representations and their similarity ranking lists (Chapter 3). In the second stage, I will review the word representations and their ranking again, from a different point of view, and train the classification trees to predict my semantic intuition about the pairs given the distributional data (Chapter 4).

Additional infomrmation on the two stages of the experiment In the first stage of the experiment, I will study the computation of the similarity ranking lists for a set of about 17 000 Finnish nouns that occur more than 100 times each in a 40 million word newspaper corpus. I will then use syntactic dependency links to select the other words, and will weight them for each word using simple co-occurrence counts. Similarity is measured by Sibson's (1969) information radius, also known as the Jensen–Shannon divergence (Lin, 1991) or, by describing the formula in words, the mean divergence to the mean. For each word, a list is made of the one hundred words that are distributionally most similar to it. This stage ends with a look at the intuitive semantic quality of a handful of pairs that could be, distributionally, some of the best.

In the second stage, I further explore the possibility of detecting semantically good or bad pairs among those that were ranked distributionally in the twenty best on their list. I use classification trees that are trained to predict my intuitive judgement of the quality of a 400-pair random sample. These classifiers use the distributional information that was available but that was under-used when the ranking lists were made.

The contents of this book The text proceeds from ta discussion of the various similarity formulas in the abstract to applying one of them to a specific corpus of Finnish newspaper data. This analysis ends with a look at those words that the computer program ranked among the twenty most similar to any word, and with an attempt to better identify the semantically related pairs among those.

This chapter provides some background. It presents the contrast between the distribution and semantics of words, keeping in mind corpus data and the practical problems of computation. The topics include:

- the distributions and meanings of words
- the similarity and semantic similarity
- the distributional hypothesis that the two are closely related
- the computation of distributional similarity

- some concerns about the real data and actually computed distributional similarities

Chapter 2 presents a number of the relevant mathematical formulas that have been presented and used when studying the distributional similarity of pairs of words. Some of the formulas are known to work better than others. I classify the formulas into three broad groups according to the mathematical representation of the data and the nature of the operations used on them: whether they treat the words as sets, vectors, or as probability distributions. I then select one formula, Sibson's (1969) information radius, to use in the computations.

Chapter 3 presents the four stages of transforming the raw corpus data into the similarity ranking lists: cleaning and parsing it, choosing the words of interest, building the computational representations of those words, and ranking the representations with respect to each other. The representations of the words and the ranking lists become the focus of further discussion in this analysis. An anecdote about a surprise with *apple* and *tax reform* sets the tone.

Chapter 4 is an experimental attempt to statistically predict whether a particularly good distributional similarity rank reflects the semantic similarity of the words in question. I have computed the similarity judgments and the ranks, and have obtained various other statistics from the word representations. I made the intuitive semantic judgments of *good* or *bad* for a random sample of 400 pairs. Decision tree classifiers were then trained with this data to determine the distributional statistics they use to predict semantic quality, and to ascertain how well they perform.

1.2 The distribution of a word

The *distribution* of a linguistic item is often defined as the contexts where the item *can* occur. I will now present three such definitions from dictionaries. First, R. L. Trask, 1993, A Dictionary of Grammatical Terms in Linguistics, defines distribution in the following way:

> **distribution** /. . . / *n.* The full range of environments in which a lexical or grammatical form can occur.

I have elided the phonetic notation. Second, P. H. Matthews, 2007, The Concise Oxford Dictionary of Linguistics, 2nd edition, offers this definition:

> **distribution** The set of contexts within sentences in which a unit or class of units can appear. E.g. the distribution of

> *hair* in written English is the set of contexts *I combed my —*, *Give me the — spray*, *My — is too long*, etc., in any of which the blank (—) can be filled by it.

Third, David Crystal, 2008, A Dictionary of Linguistics and Phonetics, 6th edition, suggests the following:

> **distribution** (*n.*) A general term used in LINGUISTICS to refer to the total set of linguistic CONTEXTS, or ENVIRONMENTS, in which a UNIT (such as a PHONEME, a MORPHEME or a WORD) can occur. Every linguistic unit, it is said, has a characteristic distribution.

This works especially well when linguists or lexicographers can make up their own data.

To describe words as people actually use them, however, a slightly different definition is needed. I must mean all those environments in which the word *does occur* in the data, and I might do well to observe the frequencies of the various items in those environments, as well. Computers can be programmed to use the variants of this notion, and linguists and lexicographers can still use everything else they know about the language, in addition to the specific data at hand.

Zellig Harris (1954) states:

> The distribution of an element will be understood as the sum of all its environments.

Harris also explicitly allows the possibility of taking frequencies into account.

1.3 The uses and meanings of a word

Isolated words are an unfortunate abstraction. The following is an occurrence of Firth's (1957) famous quip about the company words keep:

> . . . The day-to-day practice of playing language games recognizes customs and rules. It follows that a text in such established usage may contain sentences such as 'Don't be such an ass!', 'You silly ass!', 'What an ass he is!' In these examples, the word *ass* is in familiar and habitual company, commonly collocated with *you silly—*, *he is a silly—*, *don't be such an—*. You shall know a word by the company it keeps! One of the meanings of *ass* is its habitual collocation with such other words as those above quoted. . .

Firth advocates the lexicographic identification and description of such patterns. In other words, the word alone does not carry its meaning.

Corpus-based lexicography can be thought of as the clustering of concordances and as the description of the clusters. Moreover, concordances of the different-looking forms of an item should be brought together, and the different uses of similar-looking items should be put apart. This process produces the real meaning-bearing items: the words and combinations of words being used in a certain sense.

The computational approach to the vocabulary adopted here is actually a practical over-simplification. The approach can also be referred to as an approximation.

Ambiguity Semantic relations occur between words in some fixed sense. A standard example in English is the word *bank*, which can refer to an institution that deals with money, or to a formation of earth, among other things. The two meanings would have different synonyms and other related words.

Different notions of *word* are recognised in the present analysis. The most superficial is when a word is defined as a delimited sequence of letters, and occurrences of the word `bank` in written text are easy to identify even for a computer. A much deeper notion, the different instances of `bank` are identified as instances of different words, or *lexemes*, according to their context of use. Some would belong to $bank_1$ and some to $bank_2$, and so on. This disambiguation is usually easy for people to discern. Computer programs are, however, less reliable.

Even superficial analyses have their problems. For example, one might want to identify instances of `banks` or `Bank` with `bank`. Or, one might *not* want to see an instance of `bank` in `bank holiday`. Or, one might want to separate the noun instances from the verb instances.

Semantic ambiguities can be seen at different levels of granularity. For some purposes, it might be desirable to separate river banks from financial banks, but then tolerable to conflate the company and the building where the company has its offices.

1.4 Similarity

Let us turn to the different kinds of *similarity*, specifically the similarities between words. For purposes of the present analysis, similarity is considered to be a *matter of degree*. In other words, similarity is more like being *near* than being in the same spot. This means that words can be *more or less*

similar. Whereas some formulas express degrees of similarity numerically, the most interesting are those words that are the most similar.

Not an equivalence A quantified similarity is not truth-valued, so properties such as reflexivity, symmetry or transitivity do not apply. However, there are two ways to base a binary relation on it and both use a threshold. A threshold can be established based on the degree of similarity and be accepted as 'absolutely' similar, to any given word, those words whose similarity is within a fixed threshold. Another alternative is to threshold on the similarity rank, so that the absolutely similar words are among a fixed number of the most similar.

Such derivative similarity relations are not equivalence relations. For example,

1. One similarity formula, called 'confusion probability', does not even make the most similar word be the word itself. However, this is unusual.

2. Symmetry fails if one thresholds on the rank. This can be demonstrated with as little as three points on a line, placing them so that nearest to A is B but nearest to B is C:

   $$A...B.C$$

   There are also important similarity formulas that are not symmetric. For instance, skew divergence is one.

3. Transitivity fails. Repeated steps from word to word, all within the threshold, will eventually cross the threshold for the first and last word.

Not necessarily a distance Notions of similarity need not match the mathematical concept of a distance. While a distance metric can be used as a similarity formula, interesting formulas in the literature are known to break every formal property of a distance metric in the following ways:

1. It is required that similarities be compared, so that it can be determined which of two words is more similar to a third. A distance metric would express greater similarity with a smaller value, and the similarity of a word to itself with the least possible distance, 0. Some similarity formulas use the largest value for the greatest similarity.

2. Many, but not all, interesting notions of similarity are symmetric in the sense that a word would always be as similar to another as the other is to it.

3. Not all interesting similarity scores satisfy the triangle inequality. The one that is selected for the present experiments is among these.

1.5 The distributional hypothesis

Variations of the starting point I adopt have come to be known as 'the distributional hypothesis', often credited to Zellig Harris, sometimes to the quip of Firth quoted above. Magnus Sahlgren (2006) discusses Harris and suggests that his claim is based on his distributional methodology of language description. Harris does not seem to present the idea formally as a hypothesis; Rubenstein and Goodenough (1965) do so, and Miller and Charles (1991) give it a name.

Distributional and semantic similarity must not be identified with each other by definition, lest one would not be able to examine to what extent they turn out to be the same. The weaker forms of the idea are more interesting. It seems plausible that semantic and distributional similarity are related in such a way that the ability to recognize one, to some reasonable extent, would help in recognizing the other, to a lesser but still reasonable extent. My hope is to predict semantic similarity from a computational distributional similarity.

Dagan, Marcus and Markovitch (1995) credit the Linguistic String Project for an "early attempt to classify words to semantic classes" and then cite a specific passage from (Harris, 1968),

> Their work was based on Harris' *distributional hypothesis* which relates the meaning of words to their distribution relative to other words:
>
> > ... the meaning of entities, and the meaning of grammatical relations among them, is related to the restriction of combinations of these entities relative to other entities (Harris, 1968, p. 12).

Julie Weeds (2003, p. 19) cites the same passage from Zellig Harris but later on, page 23, Weeds interprets the hypothesis in a stronger form: "The tasks that we classify as *semantic* are ones which rely on the distributional hypothesis: that semantics can be predicted from syntax." James Curran (2003, p. 17) defines 'distributional hypothesis' in passing:

> Much of the existing work on synonym extraction and word clustering ... is based on the *distributional hypothesis* that *similar terms appear in similar contexts.* This hypothesis indicates a clear

> way of comparing words: by comparing the contexts in which they occur. ...

Rubenstein and Goodenough (1965) identify the hypothesis as a hypothesis, to be studied for psychological reality. From their abstract, they observe:

> Experimental corroboration was obtained for the hypothesis that the proportion of words common to the contexts of word A and to the contexts of word B is a function of the degree to which A and B are similar in meaning. ...

They therefore conclude:

> 1. The basic hypothesis investigated by the study is corroborated: there is a positive relationship between the degree of synonymy (semantic similarity) existing between a pair of words and the degree to which their contexts are similar.

A much later paper by George A. Miller and Walter G. Charles (1991) names two variants of the hypothesis:

> *Strong Contextual Hypothesis*: Two words are semantically similar to the extent that their contextual representations are similar.
>
> ...
>
> *Weak Contextual Hypothesis*: The similarity of the contextual representations of two words contributes to the semantic similarity of those words.

Both cite a statement from Zellig Harris (1954) that can be reduced to the following:

> ... we will often find that .... In other words, difference of meaning correlates with the difference of distribution.

Harris' section title is relevant: **Meaning as a function of distribution**. This idea of correlation can be adopted here, and then the question is whether the correlation is strong enough for practical computation.

A computational version of the hypothesis can also be found at least as early as 1962, when Paul L. Garvin (1962) formulated the idea of 'distributional semantics' as follows:

> Distributional semantics is predicated on the assumption that linguistic units with certain semantic similarities also share certain similarities in the relevant environments. If therefore relevant environments can be previously specified, it may be possible

> to group automatically all those linguistic units which occur in similarly definable environments, and it is assumed that these automatically produced groupings will be of semantic interest.

1.6 Semantic similarity

In this analysis, *semantically similar* words have similar meanings. Semantic similarity is an elusive notion, just as the very *meaning* of a word is elusive, but some core cases can be identified as acceptable as instances of semantic similarity. The main case I want to exclude is that where words occur in some similar-looking contexts without being close in their meanings.

Semantic relations Words can be related in their meanings in many recognised ways. Some important semantic relations have names formed with a prefix to '-onymy' (in English). Web sources say this derives from a Greek suffix -ωνυμος, in turn derived from the word ονομα, which means *name*. Four such relations give rise to core cases of semantic similarity:

1. The words can mean more or less the same. This is called *synonymy*, and the words are called *synonyms*.

2. One word can be slightly more general. This is called *hyperonymy*, with inverse *hyponymy*; the more general word is a *hyperonym* of the less general, and the less general word is a *hyponym* of the more general.

3. The words can have a slightly more general word in common. They can be called *cohyponyms* of the more general one, or possibly *sisters* or *cousins*, to use a family tree metaphor.

4. One word can mean roughly the opposite of the other. Such words are called *antonyms*. In some sense, such words share much their meaning, differing only in one polar aspect.

There are other meaning relations, such as the relation between a part and a whole, often called *meronymy*. The above four are related to the notion of similarity adopted in the present analysis.

Examples of cohyponymic similarity The names of the different kinds of fruit are semantically similar; words for the different kinds of catastrophes are semantically similar; all city names are semantically similar. However, some may be more similar than others.

Semantic hierarchy In WordNet (see (Fellbaum, 1998)), hyponymy organises English nouns into a semantic hierarchy where a few generic nodes are at the top and increasingly more specific nodes are found as one follows the links down to the hyponyms. Each node consists of a set of synonymous nouns.

This hierarchy is an indirect representation of the meanings of the English nouns in it. It also suggests a notion of semantic similarity: the short distance up and down along hyponymic links. For example, nouns could be seen as semantically similar if they appear on the same node, or if there are only a few hyponymy or hyperonymy steps between some nodes where they appear.

Philip Resnik (1998) has presented better, more sophisticated ways to compute the semantic similarity on WordNet.

Topical association Reference to the semantic relations gives some structure to semantic similarity. A different kind of semantic association between words is the association with the same topic area. I think this is too loose to be called similarity. However, this is important for information retrieval.

Intuition I do not formally define semantic similarity. However, I do take closeness in some imagined hyponymy hierarchy, as in WordNe,t as a core case of high semantic similarity. Unfortunately, for the pairs of text words, I was only able to decide if I felt intuitively good or bad about their closeness.

1.7 Computing distributional similarity

Let us refer to two words as being *distributionally similar* to the extent that their distributions overlap in a corpus. In practical computation, words are represented by simple mathematical objects that are derived from their concordances by way of drastic simplification.

Automatic linguistic analysis tools (morpho-syntactic parsers) help to identify different forms of a word and different syntactic relations between words that occur near each other. After that, everything reduces to counting the occurrences of pairs of words because similarity is quantified by a mathematical function that ultimately depends on the co-occurrence counts.

The main steps to compute distributional similarity in a given corpus are as follows:

1. Representation of each word by its concordance, simplified to the point where it can be handled

2. Use of a formula on pairs of word representations to obtain numerical similarity scores

3. Ranking the pairs to find those most similar to each given word, or clustering to find groups of similar words

The simplification of concordances involves choosing both the words themselves and some suitable parts of their concordances. The latter can be simply words that appear in the concordance, or they can also contain data from a linguistic analysis of the text.

In this analysis, I use four technical terms that help in discussing such computations: words have *attributes*, attributes have *weights*; similarity lists consist of a *head* word and its ranked *tail* words.

1. The formal object that represents a word – for the sole purpose of computing its distributional similarity with any another word – is the weighted collection of items that occurred with the word in a corpus. These items are the (computational) *attributes* of the word. Typically they are other words with or without an indication of the type of the co-occurrence.

2. The *weight* of an attribute in such a representation is a number that indicates the strength of the association between the word and the attribute.

3. I compute a similarity list for each *head word*, simply referred to as *head* for sake of brevity. The head is some word for which there is a distributional representation.

4. A similarity list itself consists of *tail words*, or simply *tails*, in the order of their decreasing similarity with the head of the list.

In the present experiment, I use a morpho-syntactic parser, and interpret all the frequent nouns in our corpus as words, and as their attributes I take the major class words (nouns, adjectives and verbs) that occur in a direct dependency relation with them. Furthermore, the name and direction of the dependency relation in the attribute are included.

For example, from the sentence `green ideas sleep furiously`, we might be able to extract the two dependency triples `idea-mod-green` (the noun `idea` has the adjective `green` as a premodifier) and `sleep-subj-idea` (the noun `idea` occurs as the subject of the verb `sleep`). Then the attributes of `idea` corresponding to an occurrence of this sentence would be `-mod-green` and `sleep-subj-`.

In this experiment, every frequent noun is the head word of its own similarity list. Every frequent noun would also be a tail word in every list, but only the hundred tails that are most similar to the head are kept. Accordingly, some of the nouns may be tails only trivially: as the most similar word to themselves but not in any other list.

1.8 Expected limits

To answer the question of to what extent one can identify the distributional similarity with semantic similarity in practice, the 'distributional similarity' is computed from the observable co-occurrence patterns in a corpus. The 'semantic similarity' refers to a loose intuitive similarity of meaning, with synonymy as a core notion, followed by relatively close hyponymy and cohyponymy and other such relations.

There are two reasons to be cautious. First, when people produce a text, which then ends up in our corpora, they do not usually have in mind the use of the text as data to illustrate the distributional characteristics of the words. Whereas a linguist and a lexicographer will be able to identify the most relevant examples, and to ignore the least relevant ones, can we program a computer to do that? Perhaps the most that can be hoped for is that the helpful types of co-occurence are more frequent than the harmful types.

Second, published examples of computed distributional similarity tend to include instances that are semantically anomalous. There are good instances, but there is no automatic method to separate the wheat from the chaff. The following examples illustrate these two points.

Natural examples are more complex than invented ones The occurrences of words in the wild are usually not meant as illustrations of the characteristics of those words or their meanings. A linguist or a lexicographer may be able to select and simplify real examples in a helpful way, or even invent new examples, but any large scale computation on corpus data must cope with whatever the corpus contains.

Table 1.1 displays three sets of English sentences that contain the word `tezgüino`, or `tesgüino`. The first is from an article by Dekang Lin (1998a). The second is his source (Nida, 1975). The third is from the wild: my own web search for the word.

Lin (1998a) suggests that the set of four example sentences might help in guessing what *tezgüino* is. His set is a simplification of an earlier set consisting of seven sentences from Eugene A. Nida (1975). These appear to be invented examples.

Table 1.1: Artificial-looking examples from Nida, and authentic examples from the web. The real examples are more complicated and less specific.

Lin	1	A bottle of TEZGÜINO is on the table.
	2	Everyone likes TEZGÜINO.
	3	TEZGÜINO makes you drunk.
	4	We make TEZGÜINO out of corn.
Nida	1	There is some *tezgüino*.
	2	A jar of TEZGÜINO is on the table.
	3	You need a lot of TEZGÜINO to get your land cleared.
	4	Everyone likes TEZGÜINO.
	5	I'll have a drink of TEZGÜINO.
	6	TEZGÜINO makes you drunk.
	7	We make TEZGÜINO out of corn, but we do not distill it.
web	1	It is hardly an exaggeration to say that almost every social activity that the Tarahumara engage in includes TESGÜINO.
	2	Different batches of TESGÜINO are said to have various qualities by the Tarahumara.
	3	The purpose was to identify size and use-wear attributes characteristic of Tarahumara TESGÜINO vessels, and how these differ from water jars, which share the similar function of holding liquids.
	4	''Rain cannot be obtained without TESGÜINO, TESGÜINO cannot be made without corn, and corn cannot grow without the rain.''

Nida offers his sentences as 'typical contexts' to "illustrate the problems involved in determining meaning directly from texts". He has in mind a human translator who is familiar with the culture where *tezgüino* is found, the Tarahumara indians in Mexico. Many of his observations are subtle. For example, the use of tezgüino for getting land cleared may not be at all obvious to those not familiar with the culture. He also notes that such examples can be misleading.

Tezgüino itself occurs in the real world: Google, the leading web search engine at the time of this writing, found many occurrences. However, the spelling of the word on the web is slightly different. Only two hits spelt it *tezgüino*, and one of them was in Spanish; the English hit directed me to the variant spelling *tesgüino*. The four `tesgüino` sentences in Table 1.1 are the first occurrence of the word in the first four documents that the search engine brought up. (Years later, Google finds many instances of Lin's example set.)

For Lin, as well as for me, the problem is that the computer has to use the available data as is, and so is basically able to count co-occurrences. So, what is it that actually happens when one uses distributional methods on actual language data? The web occurrences are the type of examples that a machine might use to represent `tezgüino`. They are more complicated than Nida's examples, but they do contain the kind of hints that Nida's examples contain. (I also found sentences that state explicitly what *tesgüino* is, so the whole corpus might have been more helpful than those four first sentences. But that would be because those web pages were about an exotic culture.)

Several arguments suggest that distributional similarity might fail to reflect similarity of meaning. The above data point to one: real observed uses vary in surprising ways. This is because people get to choose for themselves what they say and what words and structures they use, and they may choose to say odd things, even repeatedly, maybe to entertain or to annoy. This is a first argument for not thinking of distributional semantics as being trivially computable: natural language contains surprises.

I thought I once saw John Sinclair quip that nobody knows what natural language is, which is why one wants to use real examples in preference to made-up ones. I cannot find that specific reference but (Sinclair, 1996) develops the contrast between the corpus-based description of words and terminology.

Published examples include semantic anomalies Several studies show how distributional similarity is, on average, useful in the tasks relevant to meaning. It is, however, usual that some groups of distributionally similar words also contain words that are not at all similar semantically. The groups

Table 1.2: A couple of Grefenstette's (1994) similarity lists from his Appendix 3. 'Words having about the same similarity are grouped together' by the vertical bars. Apparently the group that contains `change` also contains `culture` and `tumor`, which are not shown; the part of the group shown here is all semantically anomalous.

word [*Contexts*]	*Groups of closest words*
`cell` [1156]	`tissue` | `group effect patient study change level case activity`
`tissue` [350]	`cell` | `growth cancer liver tumor` | `resistance disease lens serum`

are also likely to miss words that would have been semantically appropriate, but this is harder to notice.

Gregory Grefenstette (1994) computed similarity lists for 1 097 nouns in a medical corpus. Table 1.2 shows the words closest to the words `cell` and `tissue`, from his Appendix 3, where `cell` had the largest 'number of contexts found for it' and `tissue` was the most similar word to `cell`.

The most similar word to `cell`, `tissue`, is acceptable, and `cell`, in turn, is most similar to `tissue`. After `tissue`, however, the similarity list of `cell` is semantically strange. Nonetheless, the list of `tissue` is more promising.

Chapter 2

Some formulas for distributional similarity

The literature uses different mathematical formulas to compute similarity. Manning and Schütze (1999) present them as two groups, categorised according to the kind of mathematical objects being compared: one group of formulas for 'vectors' and one for probability distributions. However, this chapter presents the first group of similarity formulas as naturally split in two: the formulas on sets, and formulas on vectors.

1. 'Set formulas' operate on *words as sets of attributes*, which are extended to allow positive weights, using notions such as the intersection and union of sets and the size of a set. An important example is the Jaccard formula, also named after Tanimoto. With weights, Grefenstette's use of this formula can be covered.

2. 'Vector formulas' operate on elements of $\mathbf{R}^n$, using notions such as the difference of vectors, the length of a vector, and the angle between vectors. *The components of the vectors correspond to the attributes of the words.* Furthermore, important formulas include the cosine and the block distance.

3. 'Probability formulas' operate on *words as probability mass functions*, often using ideas from information theory. An important building block for these formulas is the relative entropy, also known as the Kullback–Leibler divergence.

All three types of objects assign numerical values, which are referred to here as weights, to the attributes of the words that they represent. Each type is associated with its own notation, operations, and terminology. Each type may also have restrictions on the allowed weights.

An empirical comparison (Lee, 1999) of several formulas in a disambiguation task suggests that Jaccard, L_1 norm, and Jensen–Shannon divergence are among the better ones. Lee argues that their performance is related to the emphasis they put on the shared attributes, as opposed to the attributes that only occur with one of the words. These three formulas can be seen as examples of the three different kinds: Jaccard as operating on sets, L_1 norm on vectors, and Jensen–Shannon on probability mass functions. (On the other hand, the L_1 norm as used by Lee is better interpreted as a probability formula, referred to more precisely as 'variational distance', as discussed on page 41. This means that the best group in the experiment does not contain any proper vector formula.)

Different choices of attributes, weights and similarity formula lead to different quantifications of a general notion of the similarity of words: *words are similar to the extent that they have the same attributes.* When the attributes correspond to words that occur with the word being represented, the notion of similarity is that *words are distributionally similar to the extent that they occur with the same words.*

2.1 Numerical representation of words

The calculation of distributional similarities applies to the observable co-occurrence frequencies in a particular corpus. Let us choose a set $\mathbf{W}$ of words that occur sufficiently often in the corpus, and a set $\mathbf{A}$ of attributes whose co-occurrences with each word in the corpus are counted. Then a computational representation $\breve{w}$ is set up for each word $w \in \mathbf{W}$ by assigning it a numerical weight for each attribute, so that the representation of w is a function $\breve{w} : \mathbf{A} \to \mathbf{R}$. In the sections that follow, I will interpret some such functions as set-like objects, as elements of a vector space, and as probability mass functions.

The weights are derived from the counts of occurrences in the corpus. If no co-occurrences of a word and an attribute are observed, the weight for that attribute in the representation of that word is 0. Otherwise, the weight is usually positive, and a higher weight (or a higher absolute weight) means a more significant association between the word and the attribute.

The co-occurrence frequencies need not be used as such. They may be corrected for various reasons. The logarithmic correction $x \to \log(1 + x)$ would replace the count x by its order of magnitude, which may better reflect the relative importance of the different observed counts. Moreover, if an attribute occurs particularly often in a corpus, it may occur often with some word without any particular association to that word; there are ways to take

that into account in the weight (see Curran (2003) for a discussion).

The simplest weighting scheme is to assign the constant 1 to each co-occurring attribute, completely ignoring the different frequencies. It is also possible to have negative weights, with the understanding that the absolute value indicates the strength of association. In addition, similarity formulas may require normalised weights.

Different words have a different number of attributes. In general, a more frequent word has more attributes. When the weights are normalised, the weight of an individual attribute comes to depend on the other attributes of the word.

2.2 General properties of similarity formulas

Similarity functions are variously referred to as metrics, measures, norms, scores, coefficients, divergences, or distances. Most of these are technical terms in the various fields of mathematics and may or may not apply to the formula at hand, so here the formulas will henceforth be referred to simply as formulas.

A distinction is sometimes made between similarity formulas and dissimilarity formulas. This distinction will not be made here and instead, the discussion will refer to the 'polarity' of the formula.

Some formulas require that the weights are normalised in some way. The most obvious cases are formulas that apply to probability mass functions: the sum of the attribute weights has to be 1. In any case, the weight of an attribute is meaningful mainly in comparison to the weights of other attributes.

The first aspect of a similarity formula to note is its range of values: there may or may not be upper and lower bounds. Proper distance measures are naturally bounded from below (the least possible distance is 0), but not from above; even they become bounded if the distances are between points that are confined to some bounded region of the space, typically at a fixed distance from origin.

Related to the range of the formula is its polarity: whether high values indicate high similarity and low values low similarity, or the other way around. It is easy to change the polarity formally, at least when the values are bounded. Least dissimilar words seem to be the most similar anyway, and it is this relative judgement that is of interest.

Similarity formulas can be symmetric in the sense of giving the same value for its two arguments in either order, or they can be asymmetric. The symmetric formulas are more usual. A notable asymmetric formula is Lee's (1999;

2001) skew divergence.

Almost all common similarity formulas are reflexive in the sense that they indicate perfect similarity when a word is compared to itself. Confusion probability does not have this property, yet Essen and Steinbiss (1992) do call it their 'measure of similarity' and Lee (1999) includes it in her comparison of different "measures of distributional similarity".

Nevertheless, several of the formulas fail the important triangle inequality that a distance metric has to satisfy. This comes about by two words sharing different attributes with a third, so that they are both similar with the third to some extent, but not similar to each other.

The following sections give an overview of the alternatives, grouped into the three natural families according to the operations they use:

1. The first family interprets word representations as sets of attributes. The formal starting point is the characteristic function of a set, whose range of values is then generalised. The formulas use set operations, such as union and intersection. Words are most similar when they have many shared attributes and few others.

2. The second family interprets word representations as points or vectors in a high-dimensional space. Attributes correspond to axes, or dimensions, of the space. This means that words are most similar when they are near each other in some sense.

3. The third family interprets word representations as probability distributions. This means that words are most similar when the same attributes have a high probability to occur with them.

2.3 Set formulas

In this section, I develop a way to interpret attribute weightings as 'sets' of attributes. The scare quotes are a warning that the ordinary kind of sets of attributes are a special of case of these 'sets' only to the extent that they need to be: size, subset relation, union and intersection. These "sets" are obtained by generalising the multiplicity of an element in a multiset (also known as a bag) to an arbitrary positive weight.

As always, I assume an ordinary, finite set of the attributes $\mathbf{A}$. Furthermore, in this section, I denote attribute weightings by using capital letters such as A and B, and the weight of an attribute $a \in \mathbf{A}$ in the 'set' A by $a@A$. By restricting the possible weights to 0 and 1, the weightings can be interpreted as ordinary sets: $a \in A$ corresponds to $a@A = 1$ and $a \notin A$ to

$a@A = 0$. If weights are restricted to $0, 1, 2, 3, \ldots$, the results are multisets, the weight $a@A$ being the multiplicity of a in A. For a 'set' A in general, we require only $a@A \geq 0$ for all $a \in \mathbf{A}$.

The point of the exercise is that after the appropriate definitions, it is evident that Grefenstette's weighted version of the Jaccard formula for the similarity of A and B is, indeed, the Jaccard formula:

$$\frac{\sum_{a \in \mathbf{A}} \min(a@A, a@B)}{\sum_{a \in \mathbf{A}} \max(a@A, a@B)} = \frac{|A \cap B|}{|A \cup B|} \tag{2.1}$$

Let us define the size $|A|$ of a 'set' A of attributes to be the sum of the weights $a@A$. The number of elements in an ordinary finite set of attributes is therefore a special case when seen as a 0-1-weighting:

$$|A| = \sum_{a \in \mathbf{A}} a@A \tag{2.2}$$

Let us also define a partial order for the 'sets' of attributes so that the subset relation of the ordinary sets is a special case: $A \subseteq B$ if $a@A \leq a@B$ for all $a \in \mathbf{A}$. This order is adopted to define the union $A \cup B$ and intersection $A \cap B$ so that the union and intersection of the ordinary sets are a special case:

- $A \cup B$ is the least 'set' that includes both A and B (is greater than or equal to them both, with $\subseteq$ as the order relation)
- $A \cap B$ is the greatest 'set' that is included in both A and B (is less than or equal to them both, with $\subseteq$ as the order relation)

Thus, the union and intersection are the usual least upper bound and the greatest lower bound, or the join and the meet. It is easy to see that the weighted union and intersection correspond to the maximum and minimum of the attribute weights:

$$a@(A \cup B) = \max(a@A, a@B) \tag{2.3}$$
$$a@(A \cap B) = \min(a@A, a@B) \tag{2.4}$$

Grefenstette, who uses the formula on the left side of equation 2.1 above in his dissertation (Grefenstette, 1994), calls it a 'weighted Jaccard measure'. Moreover, Charniak (1993) refers to it as a 'weighted Tanimoto measure'. It is evident that equation 2.1 holds for our 'sets' so we can accept the much simpler right-hand formula as the weighted formula, with the ordinary Jaccard formula as a special case. (Grefenstette used complicated weights,

see (Curran, 2003) for discussion. The ordinary binary-weighted Jaccard performs well in (Lee, 1999).)

It is clear that the similarity judgment $|A \cap B|/|A \cup B|$ is always in $[0\,..\,1]$, with 0 corresponding to the least similarity ($|A \cap B| = 0$, no attribute has a positive weight in both A and B) and 1 to most similarity ($A = B$, all attributes have the same weight in both A and B).

The case where $|A \cup B| = 0$ can be left undefined.

With 0-1-weights, the comparison of the attribute weightings $A = \breve{u}$ and $B = \breve{w}$, for some words u and w, separates the attributes $a \in \mathbf{A}$ into four classes: those that occur with both u and w, those that occur only with u, those that occur only with w, and those that occur with neither u nor w. Counting the attributes in each class, the following table of counts can be formulated:

	$a@B = 1$	$a@B = 0$	$\sum$
$a@A = 1$	k_1	k_2	m_1
$a@A = 0$	k_3	k_4	m_2
$\sum$	n_1	n_2	$\lvert\mathbf{A}\rvert$

There are k_1 attributes seen occurring with both u and w, and k_4 attributes seen with neither. There are k_2 attributes seen occurring with u but not with w, and k_3 attributes seen occurring with w but not with u.

Many other similarity measures can be expressed using set operations. For example, Manning and Schütze (1999) list a few, offering a brief discussion on each. They refer to these formulas as 'similarity measures for binary vectors' but recognise the set interpretation as the 'simplest way to describe a binary vector'.

$$\begin{array}{lll}
\dfrac{k_1}{\sqrt{m_1 n_1}} & \dfrac{|A \cap B|}{\sqrt{|A|\,|B|}} & \text{cosine} \\[2ex]
\dfrac{2k_1}{m_1 + n_1} & \dfrac{2|A \cap B|}{|A| + |B|} & \text{Dice} \\[2ex]
\dfrac{k_1}{k_1 + k_2 + k_3} & \dfrac{|A \cap B|}{|A \cup B|} & \text{Jaccard} \\[2ex]
k_1 & |A \cap B| & \text{matching} \\[2ex]
\dfrac{k_1}{\min(m_1, n_1)} & \dfrac{|A \cap B|}{\min(|A|, |B|)} & \text{overlap}
\end{array} \tag{2.5}$$

All contain the shared mass k_1, or $|A \cap B|$, in the numerator; most include k_2 and k_3; none contain k_4 in any way. All are obviously symmetric and all but the matching are bounded.

When expressed using the set operations, all but the cosine would seem to generalise to the weighted case with no extra work. The cosine, however, belongs more properly to the next section.

2.4 Vector formulas

In this section, I develop a way to interpret attribute weightings as the elements of a vector space over $\mathbf{R}$. The vector space is $\mathbf{R}^n$, so that the vectors are lists of n weights, where $n = |\mathbf{A}|$ and each attribute is identified by an index $k = 1, \ldots, n$. The weights need not be restricted in any way.

Three widely used similarity formulas apply to such vectors: two distance metrics (Euclidean distance and block distance, as defined below) and the cosine of the angle between the vectors. Geometrically, the two metrics correspond to the distance between two points along a straight line (Euclidean distance) and along the n axes (block distance). The angle between the vectors corresponds to the length of an arc of a unit circle; the cosine of this angle is easy to compute.

In this section, I use lower case letters such as u and w to denote the weightings themselves, subscripted to denote the weight of attribute k in the vector, u_k.

Vector spaces are defined as having two operations, which here are the multiplication of each component of a vector by a real number, and the addition of two vectors component-by-component. For $x \in \mathbf{R}$ and $u, w \in \mathbf{R}^n$, let us write $(xu)_k = xu_k$ and $(u + w)_k = u_k + w_k$ for every $k = 1, \ldots, n$.

Given these two operations, the difference of two vectors can be defined component-by-component: $u - w = u + (-1)w$. This is a key building block in the two distance metrics below.

Another key building block for the Euclidean distance and the cosine is the dot product $u \cdot w$ of two vectors u, w, which is obtained as the sum of componentwise products:

$$u \cdot w = \sum_k u_k \cdot w_k \tag{2.6}$$

The index k ranges through all the components, $k = 1, \ldots, n$. The $\cdot$ on the right denotes the ordinary multiplication of real numbers.

The dot product provides an important way to define the length of a vector $u \in \mathbf{R}^n$. This is the Euclidean norm of u, which is denoted here by $\|u\|$.

$$\|u\| = \sqrt{u \cdot u} = \sqrt{\sum_k u_k^2} \tag{2.7}$$

Using this norm, we can define the Euclidean distance between two vectors u and w, which is simply $\|u-w\|$, and the cosine of the angle between u and w.

$$\cos(u,w) = (u/\|u\|) \cdot (w/\|w\|) \tag{2.8}$$

The vectors $u/\|u\|$ and $w/\|w\|$ have the same direction as u and w but their (Euclidean) length is 1.

With another norm, which is denoted here by $|u|$, in a similar way we get the block distance $|u-w|$. This norm corresponds to measuring the distance along the axes.

$$|u| = \sum_k |u_k| \tag{2.9}$$

The vertical bars on the right denote the usual absolute value of a real number.

Distance metrics are unbounded if individual weights are unbounded. Furthermore, zero distance indicates the greatest similarity. The cosine is between -1 and 1, and the greatest similarity is indicated by 1.

The formula for the variational distance (see the discussion on page 41) of the probability mass functions is formally identical to the block distance of the vectors.

Dimension reduction methods are outside the scope of this book The vectors that are extracted from a corpus, identifying each attribute with a dimension, are very sparse in the sense that the vast majority of their components are zeroes. Some methods, however, project the data into much fewer dimensions. From the point of view adopted in this analysis, such methods appear to use the cosine on their vectors in a way that approximates the cosine on the vectors in this study. In other words, they make the relation between the individual dimensions and the actual corpus data more abstract than is the position adopted in the following chapters.

See (Sahlgren, 2005) for a discussion on random indexing, where our 'attribute' corresponds to a handful of random dimensions, and also for a discussion on 'latent semantic indexing' (or 'analysis') in which the most important dimensions are identified as the linear combinations of our 'attributes' using the singular value decomposition.

2.5 Probability formulas

This section presents the development of a way to interpret the attribute weightings as probability distributions, or more specifically, as probability

mass functions. Two building blocks of this family of similarity formulas are the means (averages) and the divergences (relative entropies, often called Kullback–Leibler divergences) of the probability mass functions. The discussion leads to the information radius formula that is adopted in the experiments of this analysis: the mean divergence to the mean.

Formally, a weighting $f_w : \mathbf{A} \to \mathbf{R}$ of the attributes of a word w is a probability mass function if no attribute has a negative weight and the sum of all attribute weights is 1. That is, $f_w(a) \geq 0$ for all $a \in \mathbf{A}$ and $\sum_{a \in \mathbf{A}} f_w(a) = 1$. For this reason, the weight of an attribute can be referred to as its probability. Furthermore, different words have different probabilities for their attributes. The less the probabilities differ, the more similar the words are.

Any non-negative and non-zero weighting can be turned into a probability mass function by dividing every weight by the sum of the weights. A simple thing to do in practice is to think of a corpus as a multiset # of word–attribute pairs, $\# : \mathbf{W} \times \mathbf{A} \to \mathbf{N}$, so that $\#(w, a)$ is the number of times the pair (w, a) was seen to occur in the corpus. Then, using a neat wildcard notation so that $\#(w, \square)$ means $\sum_{a \in \mathbf{A}} \#(w, a)$, each probability $f_w(a)$ can be defined as the relative frequency of (w, a) among those pairs where the word is w:

$$f_w(a) = \#(w, a)/\#(w, \square) \tag{2.10}$$

Appropriate relative frequencies would also provide consistent probability assignments for the pairs, $\#(w, a)/\#(\square, \square)$, for the words, $\#(w, \square)/\#(\square, \square)$ and similarly for the attributes, and for the words given the attribute, $\#(w, a)/\#(\square, a)$, but now only the f_w for every $w \in \mathbf{W}$ is needed.

The general mathematical device to represent probability distributions is a probability measure. A probability measure assigns probabilities not to individual elements, but to the sets of them. In the present case, everything is simple because I only work with finite sets of attributes. The measure μ_w that corresponds to the f_w that is used to represent the word $w \in \mathbf{W}$ is defined for any set $A \subseteq \mathbf{A}$ by simply summing up the probabilities.

$$\mu_w(A) = \sum_{a \in A} f_w(a) \tag{2.11}$$

Conversely, $f_w(a) = \mu_w(\{a\})$.

Variational distance With finite $\mathbf{A}$, any two probability mass functions $p, q : \mathbf{A} \to \mathbf{R}$ have their corresponding probability measures P, Q defined by the sum in equation 2.11, and the greatest difference of P and Q can be

stated in terms of the pointwise differences of p and q (Cover and Thomas, 1991, p. 299):

$$\max_{A \subseteq \mathbf{A}} |P(A) - Q(A)| = \frac{1}{2} \sum_{a \in \mathbf{A}} |p(a) - q(a)| \tag{2.12}$$

This is referred to as *total variation distance* or *variational distance*. It seems common to take the right-hand side, omit the factor of $\frac{1}{2}$, and still call the result *variational distance*. Jianhua Lin (1991) does just this, and (Weissman et al., 2003) begins by introducing it as 'variational, or L_1, distance'. I adopt this practice in this analysis.

Such a sum of pointwise differences is formally identical to the block distance of vectors, so the notation that I used for that can be adopted. Then the variational distance of my representations f_u and f_w of the words u and w as probability mass functions is:

$$|f_u - f_w| = \sum_{a \in \mathbf{A}} |f_u(a) - f_w(a)| \tag{2.13}$$

Lee (1999) calls this 'L_1 norm' but she also points out an interesting property that holds for probability mass functions but not for vectors in general: $|p-q|$ can be restated only in terms of the shared attributes:

$$|p - q| = 2 + \sum_{\substack{p(a)>0 \\ q(a)>0}} (|p(a) - q(a)| - p(a) - q(a)) \tag{2.14}$$

This depends on the normalisation of weights so that their sum is 1.

It is, nonetheless, common to call this variational distance of probability distributions by names which are otherwise used for the L_1 distance metric. Cover and Thomas (1991) are among those who do so. Yet they do know the variational distance as the left hand side of 2.12.

Mean probability The means of the two probability mass functions $p, q : \mathbf{A} \to \mathbf{R}$ are also probability mass functions. These can be written as $m = \alpha p + \beta q$, with the weights $\alpha, \beta \geq 0$ and $\alpha + \beta = 1$, meaning that $m(a) = \alpha p(a) + \beta q(a)$ for every $a \in \mathbf{A}$.

The specific case we used herein has the uniform weights $\alpha = \beta = \frac{1}{2}$, so that $m = (p + q)/2$ and $m(a) = (p(a) + q(a))/2$.

In the more general direction, the mean of any number of the probability mass functions $p_k : \mathbf{A} \to \mathbf{R}$ with the weights β_k is also a probability mass function, where each $\beta_k \geq 0$ and $\sum_k \beta_k = 1$. The notation looks like this:

$$m = \sum_k \beta_k p_k \text{ means } m(a) = \sum_k \beta_k p_k(a) \text{ for all } a \in \mathbf{A} \tag{2.15}$$

Confusion probability Confusion probability $p_c(u|w)$ is the mean probability of the word being u, given that the attribute is a, weighted by $f_w(a)$. To each attribute, a a probability mass function g_a can be assigned the same way f_w was assigned to each word w above:

$$g_a(w) = \#(w, a)/\#(\Box, a) \tag{2.16}$$

Next, $p_c(\cdot|w)$ can be defined as a mean of all g_a:

$$p_c(u|w) = \sum_{a \in \mathbf{A}} f_w(a) g_a(u) \tag{2.17}$$

The sum of the weights $f_w(a)$ is 1 as required because each f_w is a probability mass function.

Confusion probability does not conform to the form of the similarity formulas in this chapter: $p_c(u|w)$ cannot be stated in terms of my word representations f_u and f_w alone. It is easily restated in a form that uses f_u and f_w together with the appropriate probabilities $p(u)$ for u and $p(a)$ for all $a \in \mathbf{A}$; see (Lee, 1999).

A more interesting points is that confusion probability does not make w maximally similar to w itself. This can be seen by studying an extreme case: let a be the only attribute that occurs with w, so that $f_w(a) = 1$ and $f_w(b) = 0$ for all $b \neq a$. Then $p_c(u|w) = g_a(u)$ for all u. In particular, $p_c(w|w) = g_a(w)$, and can be the case that $g_a(u) > g_a(w)$ for some u.

Confusion probability is used in (Essen and Steinbiss, 1992) and (Grishman and Sterling, 1993).

Relative entropy Among the basic concepts in information theory are the entropy of a probability distribution and the relative entropy of two probability distributions assigned to the same values (Cover and Thomas, 1991). Relative entropy is often called the Kullback–Leibler divergence, or just the KL divergence. It is not a useful similarity formula in itself, but it is used as a building block in two interesting similarity formulas, the skew divergence and the information radius, which I will discuss below.

If p and q are two probability mass functions $\mathbf{A} \to \mathbf{R}$, and $p(a) > 0$ implies $q(a) > 0$ for all $a \in \mathbf{A}$, their relative entropy $D(p \parallel q)$ is defined as follows:

$$D(p \parallel q) = \sum_{a \in \mathbf{A}} p(a) \log \frac{p(a)}{q(a)} \tag{2.18}$$

The terms with $p(a) = 0$ are taken to be 0.

When defined, $D(p \parallel q) \geq 0$. The case $D(p \parallel q) = 0$ occurs when $p = q$. Otherwise $D(p \parallel q)$ can be arbitrarily large.

The fact that $D(p \parallel q)$ is undefined when $p(a) > 0$ and $q(a) = 0$ for some $a \in \mathbf{A}$ prevents us from using relative entropy as such, because our corpus-based weightings are sparse in the sense that most attribute weights are 0 and so $D(f_u \parallel f_w)$ would be undefined for most words u and w. A solution is to use, in the place of q, a suitable mean of p and q.

Skew divergence Having worked with the mean divergence to the mean (see information radius, below) Lillian Lee (2001) introduced another, very simple formula which she named *skew divergence*. This is the relative entropy between one of the distributions and their weighted mean. The weight α is a parameter, $0 \leq \alpha \leq 1$, so that the result is a family of measures that correct q towards p by different amounts.

$$s_\alpha(p, q) = D(q \parallel \alpha p + (1 - \alpha)q) \tag{2.19}$$

At $\alpha \approx 1$, the skew divergence is close to relative entropy, $s_\alpha(p, q) \approx D(q \parallel p)$. At $\alpha = 1$, the approximation would be exact, but this is precisely the case that needs to be avoided. At $\alpha = .5$, the skew divergence is twice one of the terms of the information radius $R(p, q)$. At $\alpha = 0$, it is no longer interesting.

With an asymmetric formula, the order of the arguments matters. Lee (2001) uses the formula to rank tail words, let us call them t, with respect to a head word, let us call it h, by $s_\alpha(h, t)$; she states that $s_\alpha(t, h)$ did not perform as well in her experiment.

Entropy Another basic quantity of information theory is entropy itself. Entropy is a property of a single probability distribution. The definition is shown here with little discussion only because entropy is used below to express various similarity measures. I will also make much use of an auxiliary function $h(x) = -x \log x$ for $x \in [0 \,..\, 1]$, which I will call 'pointwise entropy'.

$$Hp = -\sum_{a \in \mathbf{A}} p(a) \log p(a) = \sum_{a \in \mathbf{A}} h(p(a)) \tag{2.20}$$

The terms of the form $0 \log 0$ are taken as 0, which is the limit of $x \log x$ as x approaches 0 from above.

Mutual information A third quantity, mutual information, has a basic status in information entropy. In our field, mutual information tends to occur in a 'pointwise' form that has been used to measure the strength of the association between a word and an attribute. Let us adopt, merely for the duration of this paragraph, the usual $p(\cdot)$ notation for the three different

probability mass functions, with relative frequencies in our pair collection assigned as the probabilities:

$$p(w, a) = \#(w, a)/\#(\Box, \Box) \tag{2.21}$$
$$p(w) = \#(w, \Box)/\#(\Box, \Box) \tag{2.22}$$
$$p(a) = \#(\Box, a)/\#(\Box, \Box) \tag{2.23}$$

Then the (pointwise) mutual information of w and a is defined as the logarithm of the ratio of their joint probability $p(w, a)$ and the product of their individual probabilities:

$$i(w, a) = \log \frac{p(w, a)}{p(w)p(a)} \tag{2.24}$$

If this is zero, knowing that the word is w does not help to guess whether the attribute is a, and knowing that the attribute is a does not help to guess whether the word is w. Positive values indicate a frequency of co-occurrence that is higher than when w as a word and a as an attribute occur indepedently of each other. Negative values indicate a lower frequency of co-occurrence.

Hindle (1990) based his similarity formula for nouns on such pointwise mutual information values. His attributes were verbs together with the indication of whether the noun occurred as its subject or as its object. Any two nouns u and w were assigned a weight, $\mathrm{sim}(a, u, w)$, "in terms of the minimum shared cooccurrence weights" for each such attribute a. Then the similarity $\mathrm{sim}(u, w)$ of the two nouns was the sum of all such weights:

$$\mathrm{sim}(a, u, w) = \begin{cases} \min(i(a, u), i(a, w)) & \text{if } i(a, u) > 0 \text{ and } i(a, w) > 0 \\ |\max(i(a, u), i(a, w))| & \text{if } i(a, u) < 0 \text{ and } i(a, w) < 0 \\ 0 & \text{otherwise} \end{cases}$$
$$\mathrm{sim}(u, w) = \sum_{a \in \mathbf{A}} \mathrm{sim}(a, u, w)$$

(It may be that Hindle did not actually use consistent probability assignments. His paper is not quite clear on this, so his 'mutual information' may be more aptly said to be *inspired by* the real pointwise mutual information, such as the 'association ratio' of Church and Hanks (1989; 1990).)

The (average) mutual information in (Cover and Thomas, 1991) is the expectation of the pointwise mutual information. The average mutual information is defined for two jointly distributed random variables and can be usefully related to both entropy and relative entropy, among other things, but that would require some further machinery otherwise not need in this study.

Information in a statement Another important concept, which is needed in the final section of this chapter, is the concept of the amount of information in a statement. This refers, more or less, to the pointwise mutual information of the statement and that same statement.

The amount of information in a statement, $I(S) = -\log P(S)$, is essentially another way to state the probability of S, and more generally $I(S|T) = -\log P(S|T)$.

The amount of information is related to the mutual information of random variables by seeing a random variable X with values $1, \ldots, n$, say, as the set of exhaustive and exclusive statements $X = k$. Then $I(X;Y)$ is the expectation of the quantities $I(X = k, Y = j)$ that correspond to the pointwise mutual information. Finally, the self-information of a statement, say $I(X = k)$, is defined as $I(X = k, X = k)$. Entropy is average self-information.

Dekang Lin (1998b) cites (Cover and Thomas, 1991) for $I(S) = -\log P(S)$. I was unable to locate this concept of the information in a statement there, but I. J. Good (see (Good, 1983)) used related notions frequently in his system of probability-based concepts. Good is predominantly interested in 'the information about hypothesis H provided by evidence E', which he writes by using a colon, $I(H : E)$, and defines by $\log(P(H|E)/P(H))$. Self-information $I(S) = I(S : S)$ appears explicitly in (Good, 1977), parts reprinted as Chapter 23 of (Good, 1983); the paper refers to Good's 1950 book, which I have not been able to find. Pointwise mutual information is a special case of $I(H : E)$. Good (1979) credits his colleague Alan Turing for this. These notions are not cited even by authors who are familiar with Good's work. For example, Berger (1985) never mentions them.

Information radius Dagan, Lee, and Pereira (1997) used three different formulas to measure the divergence of two probability distributions: total divergence to the average, L_1 norm, and confusion probability. The first of these performed best in their experiment. In later papers (Dagan et al., 1999; Lee and Pereira, 1999; Lee, 1999, 2001), they replace the *total* divergence with the essentially equivalent *mean* divergence to the mean. They call it the *Jensen-Shannon divergence*, crediting Jianhua Lin (1991) (and a 1982 paper by Rao that I have not seen).

Lin clearly believed the measure to be new, but there is an earlier publication by Robin Sibson (1969), with a different and much more general derivation. Sibson called the formula the *information radius*. I adopt this name for the case that matches Lin's most general definition, though in the end, I only use it for two equally weighted probability mass distributions.

Sibson defines the information radius of order α for any number of arbi-

trary probability measures μ_k with weights w_k. Here I adapt his definition for $\alpha = 1$ and n probability mass functions $p_1, \ldots, p_n$ with weights $w_1, \ldots, w_n$, each $w_k \geq 0$ and $\sum_k w_k = 1$ as follows:

$$R\begin{pmatrix} p_1 & \cdots & p_n \\ w_1 & \ldots & w_n \end{pmatrix} = \sum_k w_k D(p_k \parallel \textstyle\sum_k w_k p_k) \tag{2.25}$$

His theorem 2.9 states an upper bound:

$$0 \leq R\begin{pmatrix} p_1 & \cdots & p_n \\ w_1 & \ldots & w_n \end{pmatrix} \leq \log_2 n \tag{2.26}$$

For two equally weighted probability mass functions p and q, the notation, the definition, and the bounds become simpler.

$$\begin{aligned} &R(p,q) = \tfrac{1}{2}D(p \parallel \tfrac{1}{2}p + \tfrac{1}{2}q) + \tfrac{1}{2}D(q \parallel \tfrac{1}{2}p + \tfrac{1}{2}q) \\ &0 \leq R(p,q) \leq 1. \end{aligned} \tag{2.27}$$

The general case (of order 1) matches the phrase 'mean divergence to the mean' exactly, with the divergence being relative entropy.

Lin (1991) works up from relative entropy. First, he proves that for two distributions, the total divergence to the mean can be expressed in terms of entropy, as follows:

$$D(p_1 \parallel \tfrac{1}{2}p_1 + \tfrac{1}{2}p_2) + D(p_2 \parallel \tfrac{1}{2}p_1 + \tfrac{1}{2}p_2) = 2H(\tfrac{1}{2}p_1 + \tfrac{1}{2}p_2) - (Hp_1 + Hp_2) \tag{2.28}$$

Then, he generalises to the weighted case, with $w_1, w_2 \geq 0$ and $w_1 + w_2 = 1$, arriving at the Jensen-Shannon divergence. This is also where he switches from *total* divergence to *mean* divergence. (A temporary notation $J(\ldots)$ is adopted just for the duration of this discussion.)

$$J\begin{pmatrix} p_1 & p_2 \\ w_1 & w_2 \end{pmatrix} = H(w_1 p_1 + w_2 p_2) - (w_1 H p_1 + w_2 H p_2) \tag{2.29}$$

Finally, Lin generalises from two to any number of distributions p_k with the weights w_k, each $w_k \geq 0$ and $\sum_k w_k = 1$.

$$J\begin{pmatrix} p_1 & \cdots & p_n \\ w_1 & \ldots & w_n \end{pmatrix} = H(\textstyle\sum_k w_k p_k) - \displaystyle\sum_k w_k H p_k \tag{2.30}$$

It is straightforward to derive this generalised Jensen–Shannon divergence from the information radius of order 1 for n probability mass functions with weights.

$$J\begin{pmatrix} p_1 & \cdots & p_n \\ w_1 & \ldots & w_n \end{pmatrix} = R\begin{pmatrix} p_1 & \cdots & p_n \\ w_1 & \ldots & w_n \end{pmatrix} \tag{2.31}$$

Misnomers A textbook in natural language processing (Manning and Schütze, 1999) cites the paper (Dagan et al., 1997), which found that the *total* divergence to the mean is a relatively good measure to use. Curiously, the book adopts the name 'information radius' for *that* measure, without any justification or citation. As we have just seen, a much older publication (Sibson, 1969) already used that name for *mean* divergence with good intuition and great generality. The present analysis adopts this older definition.

If the binary case of total divergence to the mean needs a geometric name, it could be referred to as the *information diameter*: twice the radius.

2.6 Pointwise radius

The symmetric, binary information radius can be worked into a form that (1) uses only shared attributes, and (2) makes explicit the contributions of individual attributes. Lee (1999) emphasises point (1); there, the formula is called the Jensen-Shannon divergence, of course. I will use point (2) when studying actual similarity judgments.

Let us set up an auxiliary function $r(x, y)$, which shall be called the *pointwise radius*, and express it in terms of the pointwise entropy function $h(x)$ in the following way:

$$\begin{aligned} r(x,y) &= -\tfrac{1}{2}(h(x+y) - h(x) - h(y)) \\ h(x) &= -x \log x \end{aligned} \tag{2.32}$$

The pointwise radius is always bounded from below: $r(x, y) \geq 0$; see section A.3 for an argument. The intended use is for the probabilities x and y of an attribute *shared* by two word representations. For those, there is also an upper bound: $r(x, y) \leq \log 2$, or simply $r(x, y) \leq 1$ when the logarithms are taken to base 2.

Restatement of the radius in terms of the pointwise radius Let us now restate the formula for $R(p, q)$ in terms of only the attributes that the words represented by p and q share. Now the radial repertoire consist of no less than *three* different expressions of the same function:

$$\begin{aligned} R(p,q) &= (D(p \parallel (p+q)/2) + D(q \parallel (p+q)/2))/2 \\ &= H((p+q)/2) - (Hp + Hq)/2 \\ &= \log 2 - \sum_{\substack{p(a)>0 \\ q(a)>0}} r(p(a), q(a)) \end{aligned} \tag{2.33}$$

With binary logarithms, $\log 2 = 1$, of course.

The contributions of other than shared attributes cancel each other out. They obviously consume some of the probability: if these were dropped from the word representations, those that remained would need to be renormalized.

2.7 Another information-theoretic formula

Dekang Lin (1998) provides another word similarity formula that does not quite fit in my three-way classification. As given in Section 5 of his paper, the formula operates on ordinary sets of attributes, yet it uses a probability weighting, and that weighting is given outside of the word representations. A slight adjustment allows me to reinterpret it as operating on a special family of weighted sets in the end of this section. In contrast to this, confusion probability could not be adjusted in the same way.

Lin proposes a kind of generic similarity formula based on the probabilities of two statements about the objects x and y being compared. The first statement 'common(x, y)' states what x and y have in common. The second statement 'description(x, y)' states what x and y are. Lin argues that the similarity of x and y should be defined as the ratio of the amount of information in these statements,

$$\mathrm{sim}(x, y) = \frac{I(\mathrm{common}(x, y))}{I(\mathrm{description}(x, y))}, \tag{2.34}$$

where $I(S) = -\log P(S)$. (Information in a statement is discussed on page 46).

Instead of spelling out the two crucial statements, Lin proceeds to replace them with suitable mathematical entities. In the application of his generic formula to the distributional similarity of words, he uses attribute sets as the descriptions of words, and uses intersections of those sets as the statements of what the words have in common. Finally, he views the attributes as being independent of each other and so is able to factor the probability of an attribute set as the product of the probabilities assigned to the individual attributes. Lin assigns the latter proportionally to the number of words that have the attribute:

$$\mathrm{sim}(u, w) = \frac{2\, I(\breve{u} \cap \breve{w})}{I(\breve{u}) + I(\breve{w})} \tag{2.35}$$

$$I(S) = -\sum_{a \in S} \log p(a) \tag{2.36}$$

$$p(a) = |\{w \in W | \#(w, a) > 0\}| / |W| \tag{2.37}$$

The numerator is to be thought of as containing the same statement about both u and w. Hence the duplication. It is important not to be misled by the notation $p(a)$. These probabilities do not sum up to 1.

Now, if Lin's starting point is to be taken seriously, it is necessary to work out the actual statements to use in his generic formula. So instead of the probabilities $p(a)$, let us consider the probability of the statement $P(a \in \breve{u})$ that a representation of a word contains the attribute. But $P(a \in \breve{u})$ has to be either 1 or 0 when u is a given word, as it is when computing the similarity of u and w. A better guess seems to be that we want $P(a \in \breve{x}|x \in W)$ where $a \in \breve{u}$. This is also a small step toward making the given information explicit in the notation, as advocated by E. T. Jaynes (2003). I therefore replace Lin's 'common(u, w)' and 'description(u, w)' with

$$\begin{array}{l} \text{common}(u, w; x, y) \\ \text{description}(u, w; x, y) \end{array} \tag{2.38}$$

that state, respectively, that each of x and y has the attributes that u and w have in common, and that x has the attributes that u has, and that y has the attributes that w has. The semicolon in the notation has no deep significance.

My reconstruction of Lin's formula for distributional similarity (and also the generic formula, with slight adjustment) becomes now the following:

$$\text{sim}(u, w) = \frac{I(\text{common}(u, w; x, y)|x, y \in W)}{I(\text{description}(u, w; x, y)|x, y \in W)} \tag{2.39}$$

I proceed to work this out, representing words x as their sets of attributes $\breve{x}$ and assuming with Lin that the presence of an attribute provides no information about the presence of any other attribute. First, let us expand the two generic statements as conjunctions of atomic statements of the form $a \in \breve{x}$. Then, let us expand $I(S|T)$ as $-\log P(S|T)$ and factor the probabilities by using the independence of the atomic statements:

$$\text{sim}(u, w) = \frac{I\big((\bigwedge_{a \in \breve{u} \cap \breve{w}} a \in \breve{x}) \wedge (\bigwedge_{a \in \breve{u} \cap \breve{w}} a \in \breve{y}) \;\big|\; x, y \in W\big)}{I\big((\bigwedge_{a \in \breve{u}} a \in \breve{x}) \wedge (\bigwedge_{a \in \breve{w}} a \in \breve{y})|x, y \in W\big)} \tag{2.40}$$

$$= \frac{2 \sum_{a \in \breve{u} \cap \breve{w}} \log P(a \in \breve{x}|x \in W)}{\sum_{a \in \breve{u}} \log P(a \in \breve{x}|x \in W) + \sum_{a \in \breve{w}} \log P(a \in \breve{x}|x \in W)} \tag{2.41}$$

The negations in the numerator and denominator cancel out each other. Incidentally, the same happens to the base of the logarithms.

Now let us identify $p(a)$ with $P(a \in \breve{x}|x \in W)$. It is clear from the longer form that Lin's probability assignment is correct for our interpretation of the formula: $p(a)$ is the proportion of the words that have a as an attribute.

The final information-theoretic form of Lin's distributional similarity formula is therefore this:

$$\mathrm{sim}(u, w) = \frac{2\sum_{a\in\breve{u}\cap\breve{w}} \log p(a)}{\sum_{a\in\breve{u}} \log p(a) + \sum_{a\in\breve{w}} \log p(a)} \tag{2.42}$$

$$p(a) = |\{w \in W | \#(w, a) > 0\}|/|W| \tag{2.43}$$

Note that this formula is not sensitive to how often the words occurred with their attributes.

Earlier in this analysis, I promised an adjustment that would put the weights in the representations of the words. The trick is simple: make $\breve{w}$ a weighted set with $a@\breve{w} = \log p(a)$ if w occurs with a, and $a@\breve{w} = 0$ otherwise. In this special family of weighted sets, $a@\breve{u} = a@\breve{w}$ if both $a@\breve{u} > 0$ and $a@\breve{w} > 0$. Therefore $a@\breve{u}$ and $a@\breve{w}$ can be replaced with $\min(a@\breve{u}, a@\breve{w})$ for the shared attributes a. I can also sum over all attributes, because for non-shared attributes, $\min(a@\breve{u}, a@\breve{w}) = 0$. Using the definitions I adopted for the weighted sets, the result is a simple notation for the formula:

$$\mathrm{sim}(u, w) = \frac{2\sum_{a\in A} \min(a@\breve{u}, a@\breve{w})}{\sum_{a\in A} a@\breve{u} + \sum_{a\in A} a@\breve{w}} = \frac{2|\breve{u} \cap \breve{w}|}{|\breve{u}| + |\breve{w}|} \tag{2.44}$$

$$a@\breve{w} = \begin{cases} \log p(a) & \text{if } \#(w, a) > 0 \\ 0 & \text{otherwise} \end{cases} \tag{2.45}$$

Lin's formula appears to be a specially weighted Dice formula.

Chapter 3

From a text corpus to a similarity ranking table of frequent nouns

In the beginning of this chapter, I have two initial resources: a body of Finnish newspaper text, and a morpho-syntactic parser of Finnish. When the chapter ends, I also have two derived resources: a collection of computational representations of the frequent nouns in the corpus, and for each frequent noun, a ranking list of a hundred of its distributionally most similar frequent nouns.

The process described here is quite standard. First, there is text in files, and that text is put through a parser. Then, the distributional information about individual words is extracted, and the pairwise similarities are computed. Then all words are ranked according to their similarity to every word. The best part of each similarity list is then kept.

Every frequent noun is assigned its own similarity lists of the frequent nouns that were distributionally most similar to it. The one word is referred to as the *head word* of the list, and the words similar to it are the *tail words* of that list. (The collection of all similarity lists can be called a similarity matrix, but its columns are not as meaningful as its rows.)

Section 3.1 describes the corpus, consisting of approximately 40 million words from the years 1995–1997 of the Finnish newspaper Helsingin Sanomat. This corpus became available for me at the Department of General Linguistics of the University of Helsinki when it was being prepared, and is now available for research purposes in the Language Bank of Finland, hosted at CSC.

Section 3.2 describes our parser, which was an early version of Connexor's functional dependency parser for Finnish. It is used to identify the words and to reduce them to base forms, and to identify the labeled dependency re-

lations between words. A newer and better version of the parser can now be used at CSC for research purposes. Publications about the parsing method and its application to English include (Järvinen and Tapanainen, 1997), (Järvinen and Tapanainen, 1998) and (Tapanainen and Järvinen, 1997).

Section 3.3 describes the vocabulary whose similarities I study: nouns, as identified by the parser, that occur in the corpus more than one hundred times, with dependency-linked verbs, nouns, and adjectives as attributes, and attribute weights simply proportional to the number of times they occur together with the word. For instance, I will look at the representation of the Finnish word for *apple* in some detail, then in less detail at the representations of those for *orange*, *potato* and *tax reform*. Finally, I will look at some of the frequency statistics of the attributes.

A step between word representations and their similarity rankings, Section 3.4 presents concrete examples of the information radii of pairs of my words in terms of the weights of their most important shared attributes. The first example concerns the Finnish words for *apple* and *orange*. Two more examples compare the word for *apple* with the words for *potato* and *tax reform*, which turned up high in the similarity list of *apple*.

Finally, in Section 3.5, I analyse the resulting ranking lists where each of my frequent nouns is followed by its hundred distributionally most similar frequent nouns, similarity being defined as the information radius of the computational representations. First I will review the top of the lists of the words for *apple* and *orange*; the former is where the *potato* and *tax reform* above come from. Then I take a random sample of 30 from the second-most frequent 10% (the 9th decile range) of the vocabulary of the frequent nouns. I will then consider the words that are ranked among the three most similar to these, and also the words that rank these among their three most similar words. I will then attempt a rough intuitive evaluation of the *semantic* similarities of these distributionally identified word pairs.

3.1 A Finnish newspaper corpus

The corpus consists of newspaper texts published in Helsingin Sanomat, a large Finnish newspaper, during the years 1995–1997. The material was gathered, in collaboration, by the Department of General Linguistics of the University of Helsinki and the Research Institute for the Languages of Finland (KOTUS). This material is now available for academic research as part of the Finnish Text Collection (ftc) in the Language Bank of Finland, hosted by CSC – the IT Center for Science Ltd (CSC). See `http://www.csc.fi` for more information.

The corpus became available to me in the Department of General Linguistics of the University of Helsinki when Mickel Grönroos was converting it to SGML according to the TEI recommendations. Thus, all work described here was done on a preliminary version of the corpus. Apart from a TEI header containing metadata about the document, each article had a body where the following occurred:

- headers, paragraphs, captions, and bylines were marked as such;
- inside a paragraph, each sentence was on its own line, in the sense familiar to those who use command line tools like `grep`;
- text contained inline markup for highlighting words

The current markup on a CSC server seems to be the same, except it is now XML. The useful feature that each sentence is on its own line seems to be undocumented or lost.

The following descriptive statistics refer to the parsed version of the corpus I prepared at the Department of Linguistics. That version of the corpus is not available to others, but CSC has an academic license for a newer version of the parser, so the exercise can be repeated there. I describe the parsing process briefly at the end of Section 3.2, page 71.

Documents Each document of the SGML form of the corpus was in its own file in a directory hierarchy by year, newspaper department code, and month of the year:

```
1995/ae/199511/hs951122akb.sgml
```

My parsed corpus mirrors that structure exactly:

```
1995/ae/199511/hs951122akb.fdg
```

The current directory structure at CSC retains all the information but is less redundant with more files in each of its directories:

```
1995/11/aehs951122akb.xml
```

There are about 30 department codes which indicate the placement of the article in the newspaper, and some of them indicate a broad topical domain for the article. For example, the `po` department deals with politics. Table 3.1 lists the department codes and document counts by year, with some sort of gloss for some of the department codes.

Code	Number of Documents				
	1995	1996	1997	$\sum$	
ac		618	125	743	Tieto&kone
ae	1 391			1 391	
ak	2 283	2 139	288	4 710	
ar		237	182	419	food
as		187	155	342	weather
at		301	226	527	science and environment
au		311	208	519	consumer
ea		306	304	610	car supplement
eb			112	112	city supplement
et	4 256	4 258	1 644	10 158	fresh
hu	5 079	4 842	1 837	11 758	persons
ka	4 885	5 561	2 760	13 206	city
kn	2 193	1 441	2	3 636	Uusimaa
ku	5 738	6 501	2 550	14 789	culture
ma[1]	2 041	1 883	826	4 750	editorial
me			25	25	travel
misc	131	684		815	
mp	3 674	3 519	1 359	8 552	opinions
nh	197	119	9	325	
po	4 110	1 883		5 993	politics
~~ro[2]~~	~~16 778~~	~~16 638~~	~~3 136~~	~~36 552~~	~~Tv programme~~
~~rt[2]~~	~~3 306~~	~~4 255~~	~~4 808~~	~~12 369~~	~~radio-TV~~
sp	13 422	13 072	6 990	33 484	sports
st	3 659	4 410	22	8 091	
ta_te	5 512	5 533	2 283	13 328	economy
tr	6 094	5 782	1 420	13 296	money
ul	7 231	7 211	2 919	17 361	foreign
va_vs[3]	762	844	967	2 573	leisure
vk	1 511			1 511	
yo	8 589	10 134	4 730	23 453	domestic
~~	~~102 842~~	~~102 669~~	~~39 887~~	~~245 398~~	~~~~
	82 758	81 776	31 943	196 477	

Table 3.1: Document counts by department and year. ([1] `ma_mn` in 1995; [2] not used; [3] `vs` in 1995)

The only use I made of these codes was to exclude the radio and TV listings from the data when collecting the frequent nouns and their concordances.

There are 245 390 documents in all, each in its own file. Of these, I use 196 469 documents. The 48 921 omitted documents are television and radio departments, `ro` and `rt`. They contain programme listings that pose problems in two ways. First, their sentence boundaries were often not well placed. This is because the material is not composed of real sentences, but of rather tabular material. Second, a linguistic matter, programme titles are often intentionally peculiar, and they keep being repeated, inflating the proportion of odd word combinations. (This might have turned out to be harmless, since the similarity statistic is based on shared attributes. I cannot know, since I did not try.)

Sentences Table 3.2 shows the number of words and sentences in each department of my parsed corpus, together with their ratios, which are a sort of average sentence length. The sentence counts are roughly the numbers of sentence end markers in the parser output. I use 4 048 668 sentences in total. The word counts are discussed below.

The average sentence length statistic in Table 3.2 is simply the ratio of the number of words and the number of sentences in the department. The unit called 'sentence' here is originally a line in a corpus file; the parser may also have added some sentence boundaries. Apart from really miss-split sentences, some of the lines are actually tabular material such as ranking lists, and these may have been split at unnatural places.

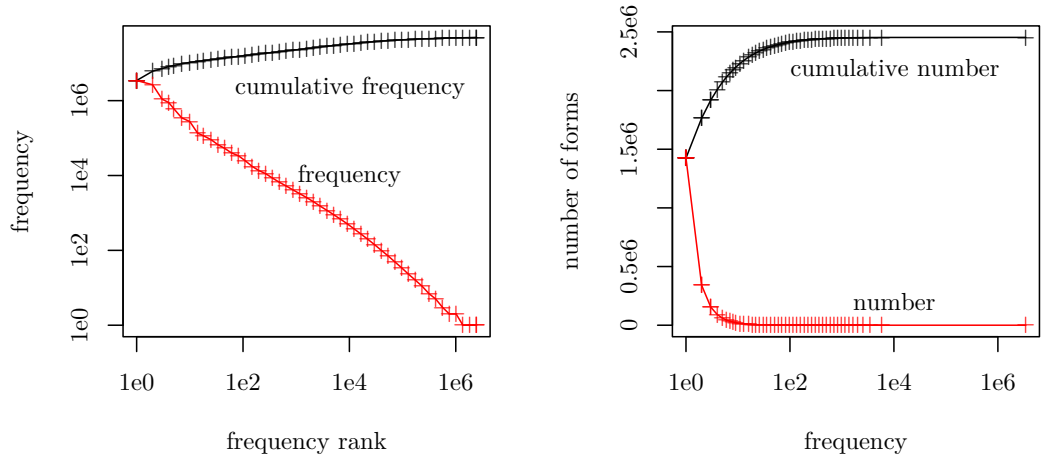
Words There are different ways to count the number of words in a corpus. The total count of token occurrences, excluding only the pseudo-token `<s>` that I added to mark sentence ends, in the part of the corpus I actually use, is 48 940 708. If only tokens that contain at least one alphanumeric character are included, so as to exclude punctuation tokens, the count is 40 561 853. This might be a reasonable figure of the size of the corpus in words. This uses the notion of a token which the parser provides.

Zipfian frequency distributions It is important to note the usual uneven distribution of the frequencies: a few of the words in every natural corpus are very frequent, whether surface forms or base forms are counted, and a large number occur only a few times.

Figure 3.1 shows the distribution of the surface-form frequencies. More than 50% of the forms occur just once. Less than 10% occur more than 10

Code	Words	Sentences	Ratio	
ac	213 609	22 300	9.6	Tieto&kone
ae	394 665	43 495	9.1	
ak	1 184 234	164 452	7.2	
ar	119 894	17 093	7.0	food
as	118 027	30 508	3.9	weather
at	129 222	10 827	11.9	
au	148 252	15 051	9.8	consumer
ea	216 594	20 320	10.7	
eb	41 204	3 905	10.6	
et	564 916	63 963	8.8	fresh
hu	1 663 034	187 680	8.9	persons
ka	2 985 417	307 892	9.7	city
kn	833 262	81 762	10.2	
ku	4 151 303	357 222	11.6	culture
ma[1]	2 247 654	183 383	12.3	editorial
me	6 112	579	10.6	travel
misc	344 529	30 894	11.2	
mp	1 906 580	167 032	11.4	
nh	105 098	10 179	10.3	
po	1 153 910	101 224	11.4	politics
~~ro[2]~~	~~4 480 012~~	~~723 765~~	~~6.2~~	~~TV programme~~
~~rt[2]~~	~~1 870 600~~	~~257 549~~	~~7.3~~	~~radio-TV~~
sp	6 587 830	670 719	9.8	sports
st	1 637 930	157 096	10.4	
ta_te	3 080 111	272 161	11.3	economy
tr	1 540 990	240 023	6.4	money
ul	3 414 584	282 031	12.1	foreign
va_vs[3]	2 531 127	250 272	10.1	leisure
vk	768 958	131 238	5.9	
yo	4 315 658	388 941	11.1	domestic
~~48 755 316~~	~~5 193 556~~	~~9.4~~		
42 404 704	4 212 242	10.1		

Table 3.2: One version of word and sentence counts by newspaper department. ([1]ma_mn in 1995; [2]ro and rt were not used; [3]vs in 1995)



Min									Median		Max
1	1	1	1	1	1	2	2	4	10	3 452 270	

Figure 3.1: The Zipfian Distribution of all surface-form frequencies in the corpus, including punctuation tokens. There are 2 453 029 distinct surface forms. Compare to the base-form frequencies in Figure 3.2, and to the noun surface-form frequencies in Figure 3.5 on page 77. (See around equation 3.1, page 60 for the making of these plots.)

times. The figure consists of three parts. In the upper-left part, the frequency of a form is plotted against the rank of frequency. In the upper-right part, the number of forms that have a given frequency is plotted against the frequency. The lower part is the list of the deciles of the frequencies of all forms.

The regularity underlying the upper-left graph in Figure 3.1 is often referred to as *Zipf's law*, for example, by Baayen (2008) and by Manning and Schütze (1999). The regularity underlying the upper-right graph is used in the Simple Good-Turing smoothing of the frequency estimates and also referred to Zipf; see Sampson (2001) (his note 6 discusses Zipf). The vertical axis of the second graph is linear, to keep the cumulative curve from being flat. The spacing on the other axes is logarithmic.

The *frequency rank* of a form indicates the position of the form in a list where the forms are placed in a decreasing order of their frequency: the most frequent form comes first and has the rank of 1, then the second most frequent form has the rank of 2, and so on. However, the relative ordering of the equally frequent forms is not of interest here.

When frequencies are plotted against their ranks, the horizontal axis is

densely populated and the frequencies decrease monotonically, though not strictly. These properties of the plot are a consequence of the artificial act of ranking.

When the numbers of items with a given frequency are plotted against the frequencies, both axes are natural and there is more variation. First, higher frequencies are sparse. This is reflected in the visible gaps between the high frequency points in the plots. Second, the numbers decrease in general, but not monotonically. This would be visible in the plots if the amount of data was small. (Perhaps these two kinds of variations are the same. Technically, the missing frequencies are associated with zeroes, after which there may come positive numbers again.)

The nth *decile* of a frequency distribution is a frequency that is equal to or higher than just $10n\%$ of the frequencies, counting equal frequencies multiply. The nth decile is also known as the $10n$th *percentile* (Moore and McCabe, 2003), or the $10n\%$ *quantile* of the distribution. The 0th decile frequency is the smallest frequency or *minimum*, or *min* for short; the 5th is called the *median*, and the 11th is the *maximum*, or *max* for short. These are roughly the eleven evenly spaced values, starting with the minimum and ending with the maximum, when the frequencies of all forms are put in their numerically increasing order.

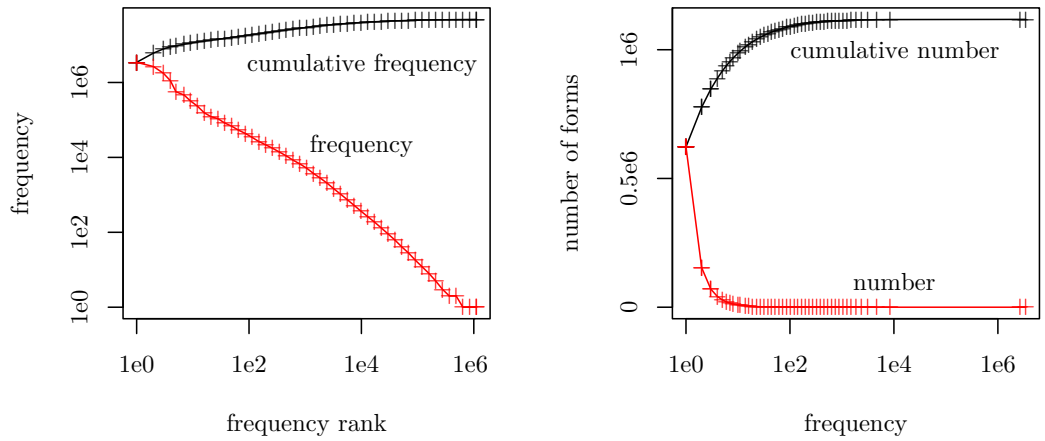
The making of the plots The plots such as Figure 3.1 in this chapter were made with R, which is both a programming language and a software package suitable for various kinds of statistical computing. The package is free software and can be found on the web at `r-project.org`. I make further use of it in Chapter 4.

The data points in both kinds of Zipfian plots are a systematic sample from the actual numerically ordered list of values on the horizontal axis. The following is a simplified R statement which produces the short and suitably spaced index sequence for the long numeric vector called `data`:

```
ix <- floor(exp(seq_len(log(length(data)))))        (3.1)
```

The logarithm of the number of actual points determines the number of points in the sample. The `seq_len` produces a vector of consecutive indices, serving as ranks on the left and as frequencies on the right, from `1` up to the given length. Exponentiation spreads these to cover the whole length of the actual data.

Distinct frequencies and their multiplicities for the upper-right plots are computed with the R function `table`, and the deciles with the `quantile`. The latter produces fractional values at times when no actual value is exactly a decile.



Min										Median			Max
1	1	1	1	1	1	2	3	5	13	3 452 270			

Figure 3.2: The Zipfian distribution of all the base-form frequencies in the corpus, including punctuation tokens. There are 1 117 734 distinct base forms. Compare these to the surface-form frequencies in Figure 3.1, and to the noun base-form frequencies in Figure 3.6 on page 78. (See around equation 3.1 on page 3.1 for the making of these plots.)

In equation 3.1 above, the resulting index sequence is assigned to the variable `ix`. Then `data[ix]` and `cumsum(data)[ix]` are plotted on the vertical axis against `ix` on the horizontal axis, with lines to connect the consecutive points.

A suitable number of data points can be selected by choosing a suitable base for the exponent function and the logarithm. In addition, all figures include the last data point, whether or not the above sampling formula would have included it.

Reduction to base forms by the parser helps only slightly; see Figure 3.2. The median frequency is still only 1, the 70% quantile grows from 2 to 3, and the 90% quantile from 10 to 13. Moreover, the proportion of small frequencies to all frequencies is still very high.

Vocabulary size One interesting aspect is the size of the vocabulary in the corpus. For this, I count the number of distinct tokens in their surface form, 2 453 029, or of distinct tokens that contain at least one alphanumeric character, 2 452 976. Nevertheless, the difference is negligible. On the other

hand, the number of distinct base forms can be counted, 1 117 734, including punctuation marks, or 1 117 701, without punctuation.

If I decide arbitrarily to study only those tokens that occur at least 10 times, the figure for the number of distinct surface forms drops to 255 117, and that for distinct base forms to 139 999, plus punctuation. A motivation for such a threshold is that the corpus as such does not tell us much about words that do not occur often enough.

The most frequent tokens Many of the most common words are grammatical in nature, such as conjunctions and auxiliary verbs. Table 3.3 shows the frequencies of the 20 most frequent tokens and base forms in the corpus.

Frequencies of the most frequent surface forms		Frequencies of the most frequent base forms	
3 452 270	.	3 452 270	.
2 655 508	,	2 655 508	,
1 098 896	ja, *and*	1 757 801	olla, *be*
891 850	on, *is*	1 107 687	ja, *and*
609 101	"	5 648 86	)
564 886	)	504 977	se, *it*
348 606	(	461 531	ei, *not*
324 664	:	348 606	(
320 387	ei, *not*	324 664	:
275 621	että	308 148	että
219 405	oli, *was*	299 095	joka
168 410	ovat, *are*	234 734	hän, *he, she*
149 796	-	205 056	vuosi, *year*
135 400	myös, *also*	161 354	myös, *also*
127 161	mutta, *but*	156 450	saada, *get*
118 927	1	154 692	tämä, *this*
116 743	ole, *is?*	149 796	-
115 492	mukaan, *according to*	144 887	mutta, *but*
112 412	kuin, *than, as*	143 825	Suomi, *Finland*
110 767	2	130 224	voida, *be able to*

Table 3.3: The 20 most frequent tokens (left) and base forms (right) in the part of the corpus that was used. Compare these with the most frequent noun tokens shown in Table 3.8 on page 76.

3.2 A functional dependency parser for Finnish

The corpus was parsed with an early version of the Connexor dependency parser for Finnish, `fi-fdg`. This parser works on running text, determining sentence and token boundaries for itself. It knows to skip simple SGML style markup, and appeared to respect the sentence boundaries marked with an `<s>` that was added in the input at the end of each line.

Method of operation The parser works by first introducing a set of possible grammatical readings for each token, then discarding token readings that are not appropriate at the place where the token appears, and by linking tokens where there is some evidence for a dependency.

The parser does not commit itself at random, except apparently for the base form when it fails to resolve a lexical ambiguity. The resulting syntactic analysis takes the form of a single partial dependency tree, with dependency links labeled by a syntactic relation that the dependent word has to its head word. The tree is often in several pieces because of missing links. Even then, I prefer to think of it as a single tree that is underspecified.

Token boundaries The plan is to collect all occurrences of the words in a corpus into concordances and to use them to represent the word. This requires concrete decisions about what it is to be a word. Some distinctions are difficult to make, some are difficult to avoid, and some are more or less a matter of choice. At the lowest level, the parser establishes token boundaries.

1. Whitespace is a good default boundary.

2. The parser may treat some whitespace as part of a token. For example, the name of the newspaper can become a single token:

   `Helsingin_Sanomat .`

3. The parser may find token boundaries even between adjacent letters. For example, in Finnish, the coordinating conjunction `mutta`, *but*, can be fused with the forms of the negative auxiliary verb `ei`, *not*. It may be desirable to insert a token boundary (the vertical bar in the example) inside such fused forms.

   `mutt|eivät .`

   A similar case in English would be the analysis of `isn't` as two tokens, `is|n't`.

4. Punctuation marks are often (not always) treated as separate tokens even though there is no whitespace between them and the adjacent tokens.

Reduction to base forms The second kind of problem occurs whenever occurrences of a word may be spelled differently. The differences may be due to variations in language, like the difference between `gray` and `grey` or between `color` and `colour` in English. More pervasive are the differences in form that are due to grammatical inflection, like the difference in number between the English `apple` (singular) and `apples` (plural). It is usual to reduce the plural forms to singular.

Finnish words occur in many inflected forms. To put these different forms together, the parser reduces them to some standard base form. For example, the forms of the English *be*, `on`, `oltiin`, `ole` and `olisinko`, and many other forms, are analysed as the verb `olla`, *be*.

The identification of upper- and lower-case letters is also an instance of such reduction. If done well, it requires the recognition of names, so that an occurrence of the company name `Apple` does not become confused with a rare capitalised occurrence of the common noun `Apple`. An example that could (perhaps) occur is: "Which flavour do you like? Apple?"

An example sentence Figure 3.3 offers an actual example of the kind of analysis that the parser produces. Figure 3.4 shows the assigned dependency (with glosses, too) of the same analysis.

In the textual form, the analysis of each token is on its own line. (I had to split a few lines to make them fit on the page here.) The lines consist of several fields:

1. a running number of the token in the sentence,
2. the actual surface form the token has in text,
3. a base form of the token,
4. a dependency link whenever the parser succeeded,
5. and a number of groups of phrase-structure and word-form labels.

Newer versions of the parser use an improved version of the same format.

The sentence can be glossed as: *The mysterious swan mentioned in the title guides a viking ship whose task is to carry a body through fire, swirls, and a narrow abyss to its shining destination.* (The swan is the swan of Tuonela in Kalevala mythology.)

```
    Surface     Base        Dependency  Phrase structure
    form        form        link        and morphology
0
1   Nimessä     nimi        loc:>2      &NH N SG INE
2   mainittu    mainita     attr:>4     &-MV V PASS PCP2 SG NOM
3   salaperäinen sala#peräinen attr:>4 &A> A SG NOM
4   joutsen     joutsen                 &NH N SG NOM
5   johdattaa   johdattaa               &-MV V ACT INF1
                                        &+MV V ACT IND PRES SG3
6   viikinkilaivaa viikinki#laiva obj:>5 &NH N SG PTV
    ,           ,
8   jonka       joka        attr:>9     &A> PRON SG GEN
9   tehtävänä   tehtävä     comp:>10    &NH N SG ESS
10  on          olla                    &+MV V ACT IND PRES SG3
11  kuljettaa   kuljettaa   subj:>10    &-MV V ACT INF1
12  ruumis      ruumis      obj:>11     &NH N SG NOM
13  tulen       tuli                    &NH N SG GEN
                                        &+MV V ACT IND PRES SG1
    ,           ,
15  pyörteiden  pyörre      cc:>18      &NH N PL GEN
16  ja          ja          cc:>18      &CC CC
17  kapean      kapea       attr:>18    &A> A SG GEN
18  kuilun      kuilu                   &NH N SG GEN
19  läpi        läpi        pm:>18      &PM PSP
20  hohtavaan   hohtaa      attr:>21    &-MV V ACT PCP1 SG ILL
                                        &A> A SG ILL
21  määränpäähänsä määrän#pää          &NH N SG ILL
    .           .
    "
24  <s>         <s>
```

Figure 3.3: A parsed sentence as text, a token per line, and long lines continued on to the next line to fit on the page. The columns are: a running number, an actual form, a base form, a dependency link, and one or more alternative taggings. The sentence is from the culture department, ku, 1995-12-13.

Major word classes: N, V, A, ADV
Minor word classes: PRON, PSP, PRE, CS, CC
Number and person: SG, SG1, SG2, SG3, PL, PL1, ...
Cases: NOM, GEN, PTV, INE, ELA, ILL, ..., ESS, TRA
Adjective degree: CMP, SUP
Voice: ACT, PASS
Mood: KOND, POT, IMP, IND
Tense: PRES, PAST
Non-finite verb forms: INF1, ..., PCP1, PCP2
Clitics: -KIN, -HAN, -KAAN

Table 3.4: Some morphological tags produced by the parser, Connexor fi-fdg. Not all tags are shown here. Only N, V and A were used; they mark the word as a noun, a verb, or an adjective.

The downward arrows in Figure 3.4 are the dependency links that connect the dependent words to their head words. In the parser output, they are assigned to the dependent word. The label on each such arrow is the name of the dependency relation. The graph is conceived of as a *single tree* with links missing, not as many small trees. (Included among the missing is the link that would have identified the root of the tree, hence the lone unlabeled node.)

Word forms Finnish word forms express a whole range of grammatical distinctions. The parser assigns to each token a sequence of labels that identify the class and form of the token. If it fails to decide between alternatives, it leaves the token with more than such label sequence. Table 3.4 lists many, not all, of the word-level labels. These correspond well with the usual grammatical categories used in the description of Finnish word forms.

Let us discuss two word form tokens, joutsen and tulen, used in the example sentence, as examples of the kinds of problems that the parser has to resolve. The analysis of the whole sentence is shown in Figure 3.3 on page 66, with the dependency structure drawn in Figure 3.4 on page 68.

The word form joutsen, *swan*, is assigned the label sequence N SG NOM. This marks it as a noun, singular as opposed to plural, and in the nominative case. The surface form is also assigned the base form of joutsen.

The word form tulen has two possible analyses in this level: it can be a singular genitive of the noun tuli, *fire*, labeled N SG GEN, or it can be a first person singular form of the verb tulla, *come*, in the active voice, indicative

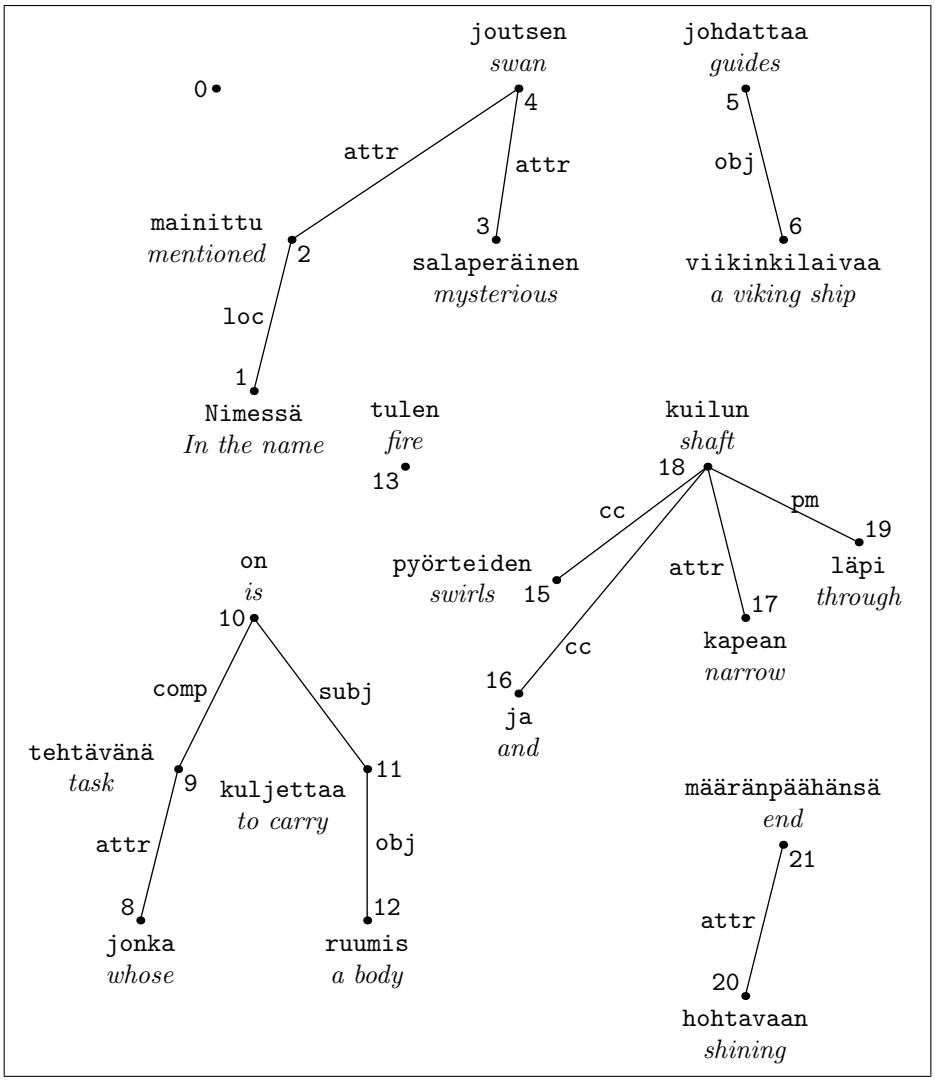


Figure 3.4: Parsed sentence drawn. There is an unlabeled root node, whereas all other nodes are labeled with the words of the sentence, and edge labels identify the dependency relations between words. Several edges are missing.

mood, present tense, labeled as `V ACT IND PRES SG1`. The phrase `tulen läpi` can be read differently in different contexts as:

`tulen läpi`, *through fire*
`tulen läpi`, *I come through*

The first reading is correct in Figure 3.3, where the token occurs as the complement of the postposition `läpi`, *through*. The postposition requires the genitive case.

Here the two readings of `tulen` correspond to different base forms, `tuli` and `tulla`. In Figure 3.3, the parser has retained both tag sequences. There the actual complement of the postposition `läpi`, *through*, is not the simple noun form `tulen` but a coordinated noun phrase `tulen, pyörteiden ja kapean kuilun`, *fire, swirls and a narrow shaft*, all in the genitive. The parser failed to decide between those two readings. In short, coordination is often difficult.

It is an actual shortcoming in the representation of the analysis that the parser had to commit to one of the base forms. In this case it happened to pick the right one, `tuli`.

Phrase structure Proceeding up from the low level of individual tokens, the parser supplies two types of relations between the tokens. The first type is a kind of a phrase structure, though indicated with labels on the tokens instead of an explicit bracketing. These labels are listed in Table 3.5; they are not used in the present study.

In the example sentence (Figure 3.3), in the phrase `salaperäinen joutsen`, *mysterious swan*, the premodifying adjective has the label `&A>`, and the head has the label `&NH`. The angle character in `&A>` is thought of as an arrow that points to the direction where the head noun is to be found.

If the parser is unable to identify a correct phrase structure label, it leaves this field empty for the token.

Dependency relations The other set of syntax labels, also shown in Table 3.5, indicates the dependency relations between individual tokens. These labels consist of two parts: the actual name of the relation and the number of the other token in the relation. The label is assigned to the *dependent* token; the other token is referred to as *the head* of that token.

In the example, the premodifying adjective `salaperäinen` and its head noun, `joutsen`, are tokens number 3 and 4 in a sentence, so the adjective is labeled by `attr:>4`. It is important to note that a phrase structure label does not identify the head word, while a dependency label does.

Dependency relations		Phrase structure roles	
`main`	main word	`&NH`	nominal head
`subj`	subject	`&+MV`	finite main verb
`obj`	object	`&-MV`	non-finite verb
`comp`	subject complement	`&ADV`	adverb
`dat`	dative object	`&A>`	premodifier
`oc`	object complement	`&QN>`	quantifier
`tmp`	time	`&AD>`	ad-adverb
`dur`	duration	`&CC`	coordinating conjunction
`man`	manner	`&CS`	subordinating conjunction
`loc`	location	`&PM`	preposition or postposition
`sou`	source		
`goa`	goal		
`qn`	quantifier		
`attr`	premodifier		
`mod`	postmodifier		
`cc`	coordination		

Table 3.5: Some phrase structure labels (right) and dependency labels (left) that the parser produces. Dependency labels are assigned to the dependent word together with the number of their head token. This gives the sentence a tree structure. Some phrase structure labels also indicate the direction where a head word is to be found in the sentence.

As I have discussed earlier, individual tokens may retain more than one word-level reading if the parser is unable to decide between them. Such tokens are often not fully linked to the tree, since the resolution of the correct form of the word is related to the resolution of its syntactic role in the sentence. In the example sentence, this is seen between the tokens 4 and 5, the noun `joutsen` and the verb `johdattaa`: the noun is the grammatical subject of the verb, but the parser has not assigned a dependency label, while the head is also left ambiguous between a finite and non-finite reading.

Ambiguity classes I will now analyse the grammatical distribution of the tokens that have the word class label `N`. Since the parser is designed to leave tokens with several analyses rather than risk a misanalysis, it is of interest to see what proportion of the tokens have only this word class label.

The counts for tokens with unambiguous classes are shown in Table 3.6.

It is possible that all readings of an ambiguous token belong to the same word class. Table 3.7 presents the counts of the different combinations of analyses in which at least one interpretation is as a noun.

Parsing the corpus For the present experiments, I made a directory hierarchy of the parsed files that exactly mirrors the hierarchy of the original corpus files. For example, the SGML form of the corpus contained this file:

`1995/ae/199511/hs951122akb.sgml`

This went to my parsing pipeline, and the result became a corresponding file in a mirror hierarchy:

`1995/ae/199511/hs951122akb.fdg`

After the parsing, I used only the parsed files as data.

Each document went through a three stage pipeline: an ad hoc correction of the markup, an extraction of the actual text, and finally the parsing.

There were some problems with the markup that remain problems. For example, a large amount of byline material remained inside the paragraphs, as if running text. This can occur at the start of a document, or at the end. In this extract from the current file `1996/05/pohs960521abs.xml`, such byline material occurs between the header and the start of the text:

18 068 372	N	noun
7 086 330	V	verb
3 743 446	NUM	number
2 593 940	A	adjective
2 405 932	ADV	adverb
1 765 825	PRON	pronoun
1 518 716	CC	coordinating conjunction
667 622	PSP	postposition
659 184	CS	subordinating conjunction
54 995	PRE	preposition
28 238	&ADV	
2 649	INTERJ	interjection
476	? N	
333	-KO	
23	GEN	
15	? V	
5	ACT	
4	-KIN	
3	PTV	
3	N ?	
1	-PA	
1	NOM	
1	-KAAN	
38 596 114		

Table 3.6: Counts of the unambiguous tokens in the parser output. (The phrase structure tag `&ADV` and the clitic tags are an error somewhere, possibly in my scripts, and the question mark `?` appears to be a spurious empty string.)

18 516 609	unambiguously N
69 314	N or V
39 104	N or A
20 847	N or ADV
11 739	N or NUM
7 284	N or ADV or one or both of A and V
7 108	N or PSP (or PRE)
5 517	N or PRON
4 371	N or A or V
4 301	N or ADV or PSP or PRE (or A or V)
4 258	N or PRON or one of V and A
854	N or one of CC and CS
696	N or NUM or PRON
564	N or some stray case or number or clitic tag
479	N or the empty tag?
425	N or INTERJ
158	N or NUM or PRON or V
119	N or V or PSP
80	N or NUM or one of A, V, ADV
1	N or ADV or PRON

Table 3.7: Counts of the ambiguity classes containing N in parser output. The vast majority of these are unambiguous.

```
</head>
<p>
        VELLAMO
</p>
<p>
        VEHKAKOSKI
</p>
<p>
        Ehdotus hallituksen tasa-arvo-ohjelmaksi on tulossa ...
        Pääministeri <hi rend="bold">Paavo Lipposen</hi> (sd) ...
```

The document metadata identifies 'Vehkakoski Vellamo' as the author of the document, so the first two paragraphs should have been one byline instead of two paragraphs. Another problem were the highlight tags in the text, also shown above, because the parser could not understand them.

As a consequence, I wrote scripts to correct such matters, often simply discarding the problematic material. Then, during the second stage, the sentences were extracted. Each physical line remaining between `<p>` and `</p>` (inside the `body` element) was to be a sentence. The extraction script added `<s>` at the end of each such line, as the parser appeared to recognise and respect it.

In the final stage, the corrected, extracted and specially terminated sentences of each document went through the parser and the result became the corresponding parsed document.

In the corpus, the used features were mainly the `<p>` and `</p>` tags and the physical separation of each sentence to its own line. Moreover, the department codes in the path names were used to exclude `ro` and `rt` from further processing.

In the parser output, the used features were:

1. token boundaries;
2. base forms;
3. the word class labels `N`, `A` and `V`;
4. dependency links, incuding relation labels.

3.3 Computational representations for frequent nouns

For the words themselves, I choose frequent nouns in their base forms, as provided by the parser; for their computational attributes I choose major class words, again in their base forms, that are in a direct dependency relation with the word, labeled by the dependency relation; and the attributes are weighted according to their relative frequencies with the word.

1. The corpus is the 1995–1997 Helsingin Sanomat material which was available at the Department of General Linguistics, parsed to obtain base forms, word classes, and syntactic dependency links.

   I exclude radio and television listings.

2. Words are the base forms of the noun tokens. I include all tokens that have `N` as a possible word class tag and occur more than 100 times, counted as their base forms.

   I excluded a number of the words for technical reasons, mostly because they never had any dependency links. Many of these were not really nouns.

3. Attributes of a word are syntactically labeled base forms of the major word class tokens with direct dependency links to the word. I include all tokens that have `N`, `A` or `V` as a possible word class tag. I use a simple notational trick to distinguish the attributes that are syntactic heads of the word from those that are its syntactic dependents.

4. Weights of attributes with respect to a word are their relative frequencies with the word. These are interpreted as the conditional probabilities of seeing an attribute with a given word. In brief, the word representation is a finite probability distribution.

3.3.1 Frequent nouns

I choose to study the similarities of nouns. For the purposes of computation, I accept as a noun every token in the parsed corpus that bears the word class label `N`. Some of these retain other word class labels, too, and some may be mislabeled, but most will be good nouns.

A study can only be conducted on those words that actually occur in my corpus, of course. Table 3.8 on page 76 lists the 20 most frequent noun surface forms and base forms.

Frequencies of the most frequent surface forms of nouns		Frequencies of the most frequent base forms of nouns	
97 815	`klo`, *o'clock*	205 056	`vuosi`, *year*
66 051	`Suomen`, *Finland's*	143 825	`Suomi`, *Finland*
57 184	`vuoden`, *year's*	118 771	`klo`, *o'clock*
48 077	`markkaa`, *mark*	89 180	`markka`, *mark*
45 588	`mk`, *markka*	69 596	`Helsinki`, *Helsinki*
44 027	`vuonna`, *in year*	60 161	`maa`, *country*
39 973	`prosenttia`, *per cent*	57 141	`mies`, *man*
38 415	`vuotta`, *year*	57 107	`prosentti`, *per cent*
32 064	`Y`	55 185	`aika`, *time*
31 292	`Helsingin`, *Helsinki's*	51 510	`asia`, *thing*
29 505	`Suomi`, *Finland*	47 403	`ihminen`, *human*
26 678	`Suomessa`, *in Finland*	46 053	`hallitus`, *government*
24 815	`A`	46 036	`mk`, *mark*
20 925	`Klo`, *o'clock*	45 199	`nainen`, *woman*
18 907	`markan`, *mark's*	44 601	`osa`, *part*
18 419	`osa`, *part*	41 697	`työ`, *work*
17 185	`Euroopan`, *Europe's*	41 427	`päivä`, *day*
16 944	`hallituksen`, *government's*	37 804	`kaupunki`, *city*
16 768	`EU:n`, *EU's*	36 323	`puoli`, *half*
16 278	`Venäjän`, *Russia's*	35 526	`loppu`, *end*

Table 3.8: Twenty most frequent surface forms (left column) and base forms (right column) of the nouns in the corpus. For an example, the forms `vuoden` and `vuotta` on the left are only some of those that combine to form the occurrences of the base form `vuosi` on the right.

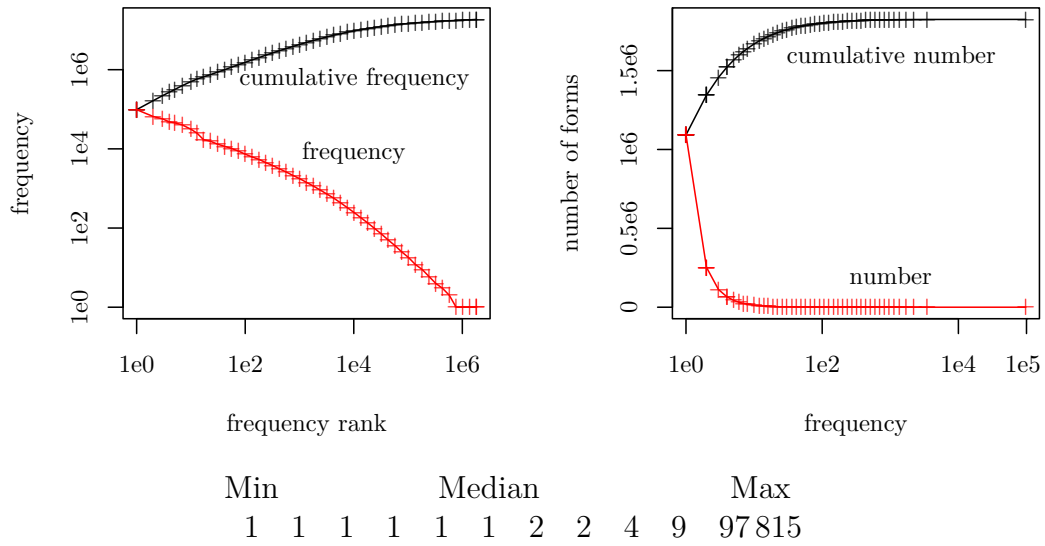
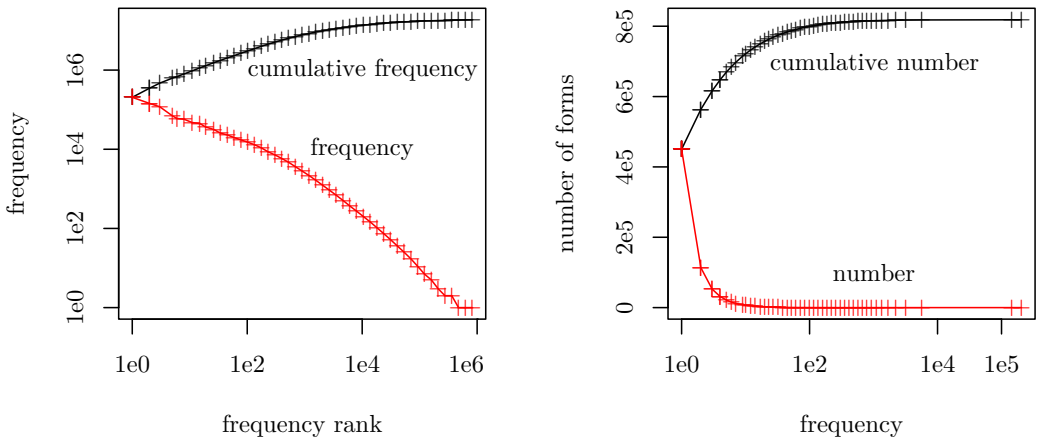


Figure 3.5: The Zipfian distribution of all the surface form frequencies of the nouns, as identified by the parser. Compare to the surface-form frequencies of all tokens in Figure 3.1 on page 59, and to the base-form frequencies of nouns in Figure 3.6. (See around equation 3.1 on page 60 for the making of these plots.)

Even for the words that do occur in the corpus, I prefer to have more than one occurrence, and I put this bar relatively high: my set of words will be those nouns that occur more than one hundred times. This count refers to the base form, as provided by the parser. Figure 3.5 on page 77 shows how the frequencies of the surface forms of nouns are distributed in the corpus. Figure 3.6 on page 78 shows the same for the base forms.

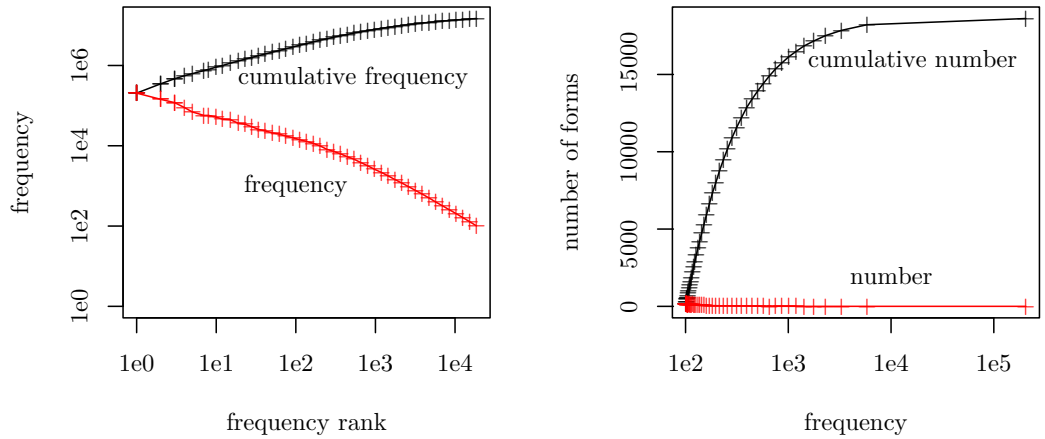
Figure 3.7 shows the frequency distribution of the frequent nouns. A number of the frequent nouns had to be excluded for technical reasons. One reason was that I used the words themselves as file names – a bad idea, which I have since abandoned, and a dozen of them contained the forward slash, which could not be used in a file name. Several were units such as `m/s`, or maybe that means *motor ship*, and one was a web address.

The other reason to exclude frequent nouns was that the parser failed to link any of their occurrences to anything, so I did not obtain the data about the word that I needed. There were 134 such tokens. Most of these are not properly nouns at all. Many are verbs. Many are not reduced to a real base form. The parser did not know what to do with them.



Min							Median			Max
1	1	1	1	1	1	2	3	5	13	205 056

Figure 3.6: The Zipfian distribution of all the base form frequencies of nouns, as identified by the parser. Compare to surface form frequencies of the nouns in Figure 3.5, to the base-form frequencies of all the tokens in Figure 3.2 on page 61, and to the base-form frequencies of the frequent nouns in Figure 3.7. (See around equation 3.1 on page 60 for the making of these plots.)



Min									Median		Max
101	115	132	155	186	229	296	410	651	1 346.4		205 056

Figure 3.7: The distribution of base-form frequencies of frequent nouns, as identified by the parser. These are the nouns that occur more than 100 times: the vocabulary of our experiments, together with the 134 words that could not be used for technical reasons. Compare to the base form frequencies of all nouns in Figure 3.5. (See around equation 3.1 on page 60 for the making of these plots.)

After the exclusions, I am left with 17 835 distinct nouns with over 100 occurrences in the corpus and at least 1 dependency link to at least 1 of the occurrences. These are my data.

3.3.2 Dependency-linked computational attributes

The next step was to build the actual computational representations of the chosen vocabulary of the frequent nouns. In this book, the (computational) attributes of a word consist of those nouns, adjectives, and verbs (in base form) that the parser analyses as occurring in a dependency relation with the word, together with the relation label that the parser assigns to the dependency link. The weight of an attribute is its relative frequency of such an occurrence with the word.

We start with a partial concordance of `omena`, *apple*, to get a glimpse at the origin of such attributes and their weights in the corpus. Then we record four word representations to be used as examples. The first two, `omena`, *apple*, and `appelsiini`, *orange*, are an arbitrary choice of two words that might be interesting to compare. The second two, `peruna`, *potato*, and `vero#uudistus`, *tax reform*, turned out to be distributionally relatively similar to `omena`. One of them is also semantically somewhat similar to `omena`, and the other is not in any way semantically similar to `omena`.

Finally, this section presents statistics on the number of attributes the words have, and on the number of words with which the attributes occur.

Concordance Table 3.9 shows some of the sentences in the corpus that contain the word `omena` *apple*. The sentences are separated with `<s>`, just as they are in the actual corpus files. The tokens separated by a space come from the parser; the one split word is an error in the corpus itself. The form of the word itself is in small caps: `OMENIA`. Some other words are slanted – *`vihreitä`* – to show that they are used as computational attributes of the word; see the next section for more information.

Table 3.10 shows the attributes of `omena`, *apple*, from Figure 3.9.

A first example of a word representation Table 3.11 shows the attributes in our representation of `omena`, *apple*. We get the probabilities by simply dividing the frequency of occurrences with an attribute by the total number of occurrences with any attribute. The high probability of the verb *be* is due to its high frequency alone; there is no reason to believe that it has any particular association to *apple*. The high probability of *green*, however, is an accidental property of our data: many occurrences are not a natural

ACHEn performanssissa Sankarin katalogi tai 229 väärää liikettä tarvitaan mm. *vihreitä* OMENIA , *paperia* , teippiä , narua , asetonia , kynttilöitä , tulitikkuja , sikareita , savukkeita , olutta , maitoa , coca-colaa , jogurttia , ketsuppia , sinappia ja samppanjaa . . . <s> Saksalaismiehityksen aikana hän välttyi niukin naukin teloitukselta OMENOIDEN *varastamisesta* . <s> Kalliossa Eläintarhan huvilassa toimiva Nukketeatteri *Vihreä* OMENA *juhlii* parhaillaan 25. vuottaan Gösta Kjellinin kirjoittamalla , yli 3-vuotiaille suunnatulla näytelmällä Mimmi Mamma Mummu . Varsinaiset juhlaviikot ovat syyskuussa . <s> Mimmi Mamma Mummu ei yllä Saapasjalkakissan , edellisen *Vihreältä* OMENALTA *näkemäni* esityksen tasolle . <s> " Kaupoista tuli valoisia itsepalvelumyymälöitä , joissa OMENAT ja *appelsiinit* kasattiin kauniisiin kekoihin . <s> Runoissa kertautuvat tutut kuvat : syreenit , seetrit , *korpit* , OMENAT , *kivet* , siemenet ja pilvet . <s> Lumikissa *oleellinen* OMENA *varastetaan* monta kertaa , ja myrkyllinen palanen joutuu lopulta hoitajan kurkkuun . <s> SAMOIN ihmetteli uusi opettaja OMENIEN määrää , kun kymiläiset lapset pyysivät koulusta lomaa päästäkseen OMENAN *ottoh* auttamaan vanhempiaan . <s> Merkityksen muuttuminen heijastuu kielestä siten , että paikoin murteissa perunalle annettiin selventävä nimitys maaperuna ja päärynälle puuperuna - siis samoin kuin OMENA *-sanallekin* . <s> T uotan sävellyksiä niin kuin omenapuu OMENIA , luonnehti Saint-Saëns itseään . <s> Tärkeilemättömään olotilaan kehottaa myös kottikärryllinen OMENOITA , jotka *eivät* ole katseltavaksi vaan syötäväksi . <s> Nyt hänen Lucanderin eteen tekemänsä työ alkaa *tuottaa* OMENIA omaankin koppaan . <s> Newton istui puun alle ja *sai* OMENAN päähänsä - miltä maailmankuva nyt näyttäisi , jos hän olisi istahtanut vaikkapa muurahaispesään , Bisquit kysyy . <s> Kirjanomistajamerkissä saa leikkiä vaikkapa sukunimellä : Mansikan merkissä on *mansikka* , *Omenamäen* OMENA , Kuuttisella hylje , Kärjellä palokärki .

Table 3.9: Partial concordance of OMENA, *apple*, in the 1996/ku part of the corpus, tokenised by the parser, KEYWORD and the *attributes* emphasised. These are surface forms, as in a running text, so the corresponding base forms and dependency links used in the representation of the word are not shown.

$f(a)$	a	Glosses of uses
4	`-attr-vihreä`	***green*** *apples*; ***Vihreä*** *Omena*
2	`varastaa-obj-`	*for* ***stealing*** *of apples*; *apple is* ***stolen***
1	`tuottaa-obj-`	*begins to* ***produce*** *apples*
1	`-#sana-attr-`	*the* ***word*** *omena*
1	`saada-obj-`	*Newton* ***got*** *an apple in his head*
1	`ottoh-attr-`	
1	`nähdä-sou-`	*the last play from Vihreä Omena that I* ***saw***
1	`-mod-ei`	*apples that are* ***not*** *for watching*
1	`mansikka-cc-`	*Mansikka's sign has a* ***strawberry***
		Omenamäki's an apple
1	`korppi-cc-`	
1	`juhlia-subj-`	*Vihreä Omena* ***celebrates*** *its 25th year*
1	`-cc-paperi`	
1	`-cc-kivi`	
1	`-cc-appelsiini`	
1	`-attr-oleellinen`	*Snow White's* ***essential*** *apple*
1	`-attr-omena#mäki`	

Table 3.10: Attributes of `omena` from the concordance in Figure 3.9, with counts and glosses for the containing expressions. Table 3.11 displays the final representation of `omena` based on the full concordance.

$p(a)$	$f(a)$	a
0.3083	255	`-attr-vihreä`, *green*
0.0193	16	`olla-subj-`, *be*
0.0157	13	`-attr-kotimainen`, *domestic*
0.0133	11	`kuoria-obj-`, *peel*
0.0121	10	`syödä-obj-`, *eat*
0.0085	7	`-attr-iso`, *big*
0.0085	7	`kilo-mod-`, *kilo*
0.0085	7	`päärynä-cc-`, *pear*
0.0073	6	`myydä-obj-`, *sell*
0.0060	5	`-attr-ulko#mainen`, *foreign*
0.0060	5	`malmi#talo-attr-`, *Malmi house*
0.0060	5	`ei-subj-`, *not*
⋮		
0.0012	1	`tähti#näyttelijä#käsi#nukke-attr-` *star actor hand puppet*
1.000	827	

Table 3.11: Attributes of the word `omena`, *apple*, ranked by their weight $p(a)$ with the word. Each a occurs $f(a)$ times with the word; $p(a) = f(a)/827$. The word has 387 attributes, and 309 (80%) occur with it just once. Table 3.10 glosses some concrete occurrences of the attributes with `omena`.

composition of *green* with `apple`, but come instead from the occurrence of the name of a puppet theatre group, as seen in the concordance sample of Table 3.9 which is taken from the culture department of the newspaper.

A second example of a word representation Table 3.12 lists the most important attributes of the word `appelsiini`, *orange*.

A third example of a word representation Table 3.13 lists the most important attributes of the word `peruna`, *potato*. In the present analysis, this is distributionally the second most similar word to `omena`. It is also interpreted as being semantically similar to it.

And a fourth example of a word representation Table 3.14 lists the most important attributes of the word `vero#uudistus`, *tax reform*. This word is also distributionally relatively similar to `omena` when we get that far

$p(a)$	$f(a)$	a
0.0424	7	`raastaa-subj-`
0.0364	6	`kuori-attr-`
0.0303	5	`mehu-attr-`, *juice*
0.0303	5	`-attr-kello#peli`, *clockwork*
0.0242	4	`sitruuna-cc-`, *lemon*
0.0242	4	`ei-subj-`, *not*
0.0242	4	`-cc-sitruuna`, *lemon*
0.0182	3	`tomaatti-cc-`, *tomato*
0.0182	3	`olla-subj-`, *be*
0.0182	3	`kuoria-obj-`, *peel*
0.0182	3	`banaani-cc-`, *banana*
0.0182	3	`-cc-banaani`, *banana*
⋮	⋮	
0.0061	1	`-#näyttämö#esitys-attr-`
1.0000	165	

Table 3.12: Attributes of the word `appelsiini`, *orange*, ranked by their weight $p(a)$ with the word. Each a occurs $f(a)$ times with the word; $p(a) = f(a)/165$. The word has 119 attributes, and 99 (83%) occur with it once.

$p(a)$	$f(a)$	a
0.0296	41	`olla-subj-`, *be*
0.0289	40	`-attr-keittää`, *cook*
0.0238	33	`-attr-uusi`, *new*
0.0224	31	`-attr-kuuma`, *hot*
0.0173	24	`kuoria-obj-`, *peel*
0.0173	24	`keittää-obj-`, *cook*
0.0144	20	`-attr-ranskalainen`, *French*
0.0137	19	`-cc-porkkana`, *carrot*
0.0130	18	`syödä-obj-`, *eat*
0.0108	15	`g-mod-`, *g*
0.0108	15	`-cc-sipuli`, *onion*
0.0094	13	`kilo-mod-`, *kilo*
⋮	⋮	
0.0007	1	`-#kala-cc-`, *fish*
1.0000	1 384	

Table 3.13: Attributes of the word `peruna`, *potato*, ranked by their weight $p(a)$ with the word. Each a occurs $f(a)$ times with the word; $p(a) = f(a)/1\,384$. The word has 693 attributes, and 518 (75%) occur with it just once.

$p(a)$	$f(a)$	a
0.0921	21	`-attr-suuri`, *great*
0.0658	15	`-attr-vihreä`, *green*
0.0482	11	`olla-subj-`, *be*
0.0307	7	`toteuttaa-obj-`
0.0219	5	`-attr-ekologinen`, *ecological*
0.0175	4	`tarvita-obj-`, *need*
0.0175	4	`olla-loc-`, *be*
0.0175	4	`-attr-vuosi`, *year*
0.0132	3	`yhteys-attr-`, *connection*
0.0132	3	`tehdä-obj-`, *do, make*
0.0132	3	`neuvottelu-mod-`, *negotiation*
⋮	⋮	
0.0044	1	`-attr-ajan#kohtainen`, *current*
0.0044	1	`-attr-Ruotsi`, *Sweden*
1.0000	228	

Table 3.14: Attributes of the word `vero#uudistus`, *tax reform*, ranked by their weight $p(a)$ with the word. Each a occurs $f(a)$ times with the word; $p(a) = f(a)/228$. The word has 140 attributes, and 112 (80%) occur with it just once.

in our studies, but it is not semantically similar to it. This case turns out to be pathological.

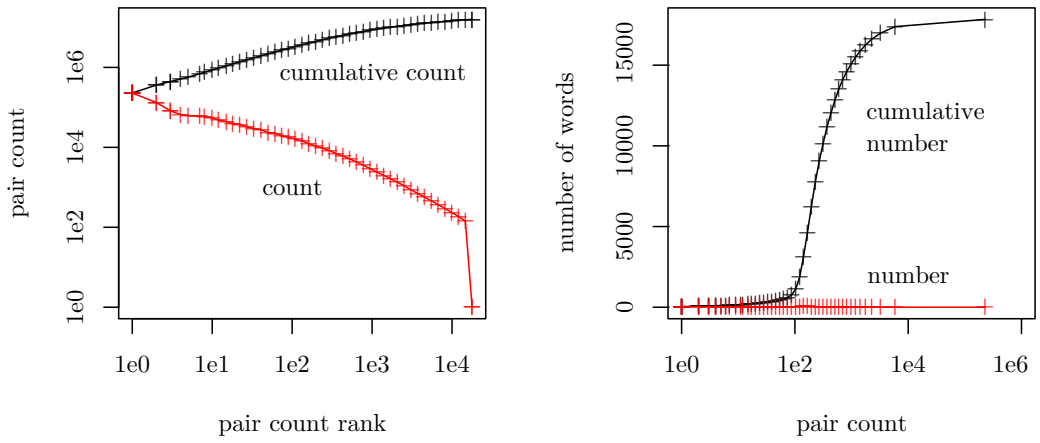
How many attributes does a word have Figure 3.8 shows the numbers of attribute occurrences with the words. This is not quite the same as the number of times the word itself occurs in the corpus: unlinked occurrences are not counted here, and multiply linked occurrences are counted multiply.

Figure 3.9 shows the numbers of the different attributes that words have. The words with very few attributes are all in the lowest decile range. Table 3.15 lists a random sample of those with at most 10 distinct attributes. There were 290 in total.

How many words have an attribute I turn the concordances of my 17 835 nouns into the computational representations of those nouns. For this purpose, I consider just the major class words that the parser links directly to the noun in question. By 'major class' I refer to nouns, adjectives, and verbs;

Word	Number of attributes	Number of pairs
9#-	5	7
BE	2	6
Noe	3	553
PSP-#prime	2	2
Suomen_Helasto	3	273
ajaa	4	5
hallinto#katu	5	5
kp	5	37
meri#sää	7	12
optimal	10	20
paasi#vuoren#katu	1	1
päässä	2	2
secura	2	352
sportmagasinet	6	7
takana	1	1
tasa#tunti	3	33
teht	1	352
toistaa	2	2
tyytymä	5	279
uutis#ikkuna	1	1

Table 3.15: A sample from the 290 nouns with at most 10 distinct attributes. Each had a 5% chance of inclusion in the sample. The first of the two numbers is the number of attributes the noun has, and the second is the number of its occurrences with an attribute.



Min					Median					Max
1	120	149	180	215	265	343	473	737	1 562.6	228 983

Figure 3.8: Numbers of times a word occurs with an attribute. Compare these to the numbers of attributes the words have, Figure 3.9. (See around equation 3.1, page 60 for the making of these plots.)

these in turn are merely the tokens that retain at least one reading with one of the word class labels `N`, `A` and `V`. All these, in their base form, together with the label of that dependency relation, become the computational attributes of the noun. There are 829 341 distinct attributes in total, most of which occur with one word only and therefore do no useful similarity work.

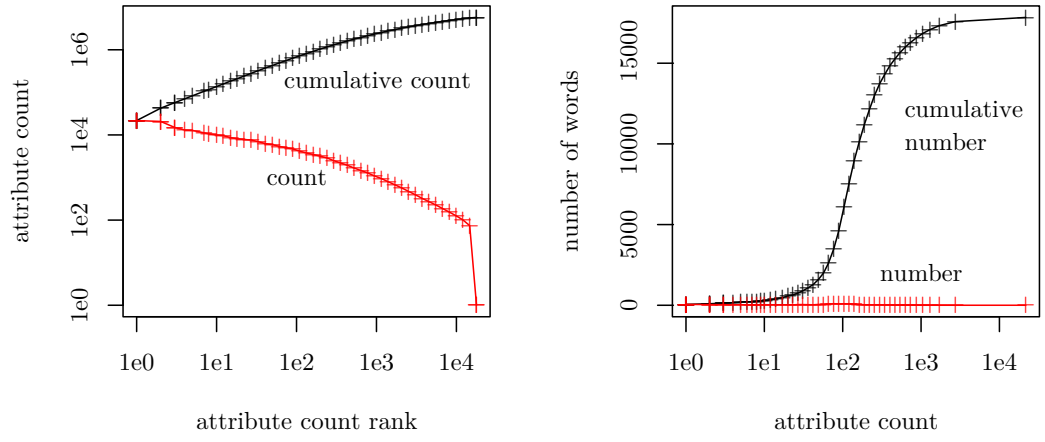
Figure 3.10 shows the number of times that an attribute occurs with some word. Again, less than 10% of what I used turn out to be useful.

Figure 3.11 shows the distribution of the number of distinct words for the attributes. Over 60% of our attributes occur with only one word, and are thus of dubious value. Under 20%, or rather under 10%, occur with sufficiently many words to be of any real interest. (I was unaware of this when I computed the similarities.)

Unfortunately, I did not have these numbers until long after I had made the computations. However, 10% of 800 000 is 80 000, so there are still many attributes that do work.

Table 3.16 lists the 10 attributes that occur with most words. Almost all of them involve the verb `olla` *be*, and the rest involve other very general verbs.

Table 3.17 lists some attributes that occur with 18 distinct words. This

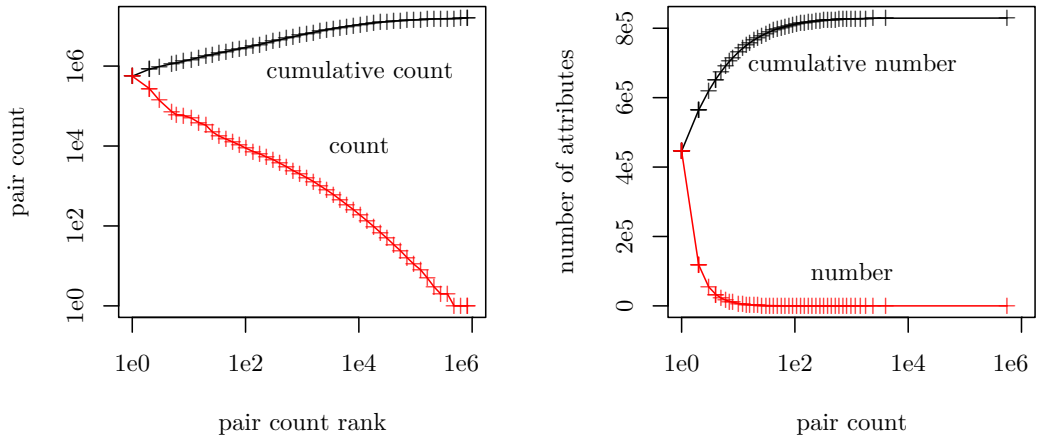


Min					Median					Max
1	53.4	78	97	116	140	176	233	340	645	21 738

Figure 3.9: The numbers of attributes the words have. Compare these to the times of occurrence with any attribute, Figure 3.8. (See around equation 3.1 on page 60 for the making of these plots.)

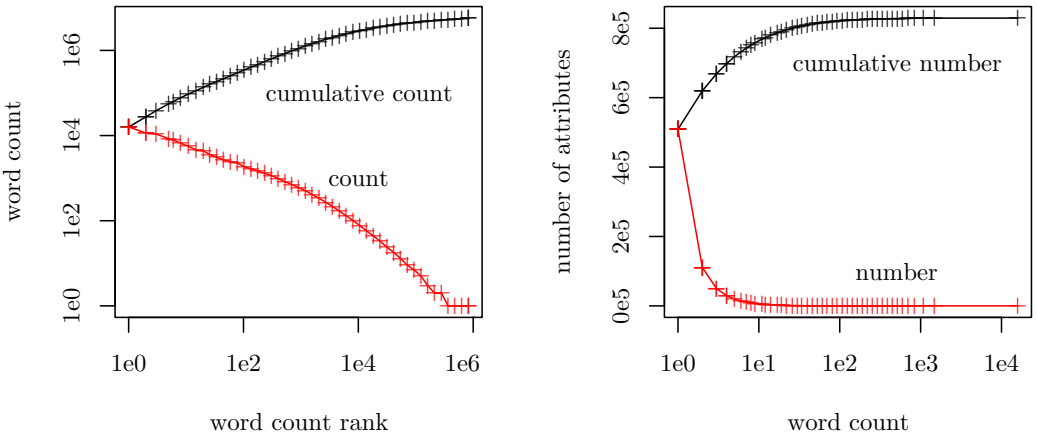
Attribute	Number of words	Number of pairs
`olla-subj-`, *be*	15 929	558 660
`olla-loc-`, *be*	11 448	274 747
`ei-subj-`, *not*	10 870	97 025
`olla-comp-`, *be*	9 037	139 778
`olla-sou-`, *be*	8 411	72 105
`-mod-olla`, *be*	7 826	37 579
`olla-obj-`, *be*	6 928	58 055
`saada-subj-`, *get*	6 623	32 346
`olla-goa-`, *be*	6 183	35 586
`tulla-subj-`, *come, become*	5 651	23 299

Table 3.16: Attributes that occur with the most words. The first number is the count of the distinct words. The second number is the count of the occurrences.



Min				Median						Max
1	1	1	1	1	1	2	3	5	13	558 660

Figure 3.10: The number of times the attributes occur with any word. Compare to the number of words with which the attributes occur, Figure 3.11, and to the number of times the words occur with any attribute, Figure 3.8. (See around equation 3.1 on page 60 for the making of these plots.)



Min									Median		Max
1	1	1	1	1	1	1	2	3	8	15	929

Figure 3.11: Distribution of the number of words for attributes. Compare these to the numbers of occurrence with any word, Figure 3.10, and to the numbers of attributes of words, Figure 3.9. (See around equation 3.1 on page 60 for the making of these plots.)

Attribute	Number of words	Number of pairs
`lumi#sade-attr-`, *snowing*	18	24
`-mod-alistaa`	18	26
`-attr-murskaava`, *crushing*	18	29
`-attr-Pen`	18	22
`-attr-Komonen`	18	20
`-attr-Jessica`	18	105
`-attr-Alexandra`	18	31

Table 3.17: A random sample of the 1 997 attributes that occur with 18 distinct words, each with 0.5% chance of inclusion.

is the 95% quantile.

Table 3.18 lists some attributes that occur with 8 distinct words. This is the 90% quantile.

Attribute	Number of words	Number of pairs
`kehitys#avu-cc-`, *aid*	8	8
`aneemisuus-attr-`, *anaemia*	8	9
`-attr-venttiili`, *valve*	8	9
`-attr-laji#tyypillinen`, *typical to species*	8	10
`-attr-kauppa#tieteellinen`	8	17
`-attr-jännitys#näytelmä`	8	10
`-attr-eristyä`, *get-isolated*	8	9
`-attr-Fisher`	8	12

Table 3.18: A random sample from the 8 548 attributes that occur with 8 distinct words – the 90% quantile. Each such attribute had a 0.1% chance of inclusion.

3.4 Computing the similarity of a pair of nouns

Let us now turn to look closer at the computation of the information radius of a word pair. Recalling the formula from Section 2.5, let us proceed to study three concrete examples.

Computing the information radii From among all the different similarity formulas, I choose one: *information radius* (Sibson, 1969), also called the Jensen-Shannon divergence (Lin, 1991) or something like the 'mean divergence to the mean'. This formula was in the best group in Lillian Lee's comparison study (Lee, 1999), together with the binary Jaccard and the block distance for probability distributions. (She uses L_1 terminology for the block distance. I would now call it variational distance when it is applied to probabilities).

This formula has been seen in three different forms, repeated here:

$$\begin{aligned} R(p,q) &= (D(p \parallel (p+q)/2) + D(q \parallel (p+q)/2))/2 \\ &= H((p+q)/2) - (Hp + Hq)/2 \\ &= 1 - \sum_{\substack{p(a)>0 \\ q(a)>0}} r(p(a), q(a)) \end{aligned} \tag{3.2}$$

The constant 1 in the last line is $\log 2$. The auxiliary $r(x, y)$ here is the *pointwise radius* from Section 2.6, here again in terms of the pointwise entropy $h(x)$:

$$\begin{aligned} r(x,y) &= -\tfrac{1}{2}(h(x+y) - h(x) - h(y)) \\ h(x) &= -x \log x \end{aligned} \tag{3.3}$$

Again, the logarithm in $h(x)$ here is binary; another base would work fine if accompanied by the corresponding adjustments to a few constants. The formula is bounded:

$$0 \leq R(p,q) \leq 1$$

It is important to also recall the bounds and their meaning: 0 corresponds to greatest similarity, 1 to the least, as if measuring a distance.

Three example pairs of words I re-examine the four example words from the previous section, now in pairs: we see details about the computation of the similarity of the word `omena`, *apple*, with each of the three other words `appelsiini`, *orange*, `peruna`, *potato*, and `vero#uudistus`, *tax reform*, respectively. The first pair is somewhat random, in the sense that I chose it without looking at the data. The other two words, `peruna` and

vero#uudistus, appear relatively high on the similarity list of omena. The first of them is intuitively acceptable, semantically, though certainly not synonymous with omena, while the other is wildly inappropriate.

Each example is a table that shows the weights of those attributes that are shared by the pair of words, ranked by their pointwise radius. These can be compared to the attributes important to the two words themselves, shown in the previous section. Captions of the tables in this section also give the total number of shared attributes and their proportion to the total number of attributes in the words.

The first example pair of words Table 3.19 shows some of the attributes shared between the two example words omena, *apple*, and appelsiini, *orange*. The attributes are ranked by their pointwise radius $r(p, q)$, where p stands for the weight of the attribute in the representation of omena, and q for the weight in the representation of appelsiini. From the sum 0.1645 of the pointwise radii of the shared attributes, the information radius is $1 - 0.1645 = 0.8355$, rather far from 0.

As was evident from table 3.11, omena had the single most important attribute -attr-vihreä, *green*, with as much as 30% of its probability mass. This attribute is not important in the comparison of apple and orange. In fact, it does not appear in Table 3.19 at all.

The most important shared attributes in Table 3.19 are the uninformative verb olla, *be*, and the clearly appropriate verbs kuoria, *peel*, and syödä, *eat*. Note that kuoria, *peel*, occurs in at least two different roles with both words: kuoria-obj- and -attr-kuoria. Some other important shared attributes refer to the size of the fruit, other kinds of fruit (or berries) mentioned in coordination with the main words, and buying (or less informatively, taking) the fruit. The uninformative negative verb ei also occurs.

The second example pair of words Table 3.20 lists the most important attributes that omena shares with peruna.

The third example pair of words Table 3.21 shows what happened between omena and verouudistus. These words share only 14 attributes. One of those, -attr-green, *vihreä*, is by far the most important attribute of omena. That attribute is also relatively important to verouudistus. Moreover, most of the other shared attributes are rather uninformative. Finally, verouudistus has far fewer attributes than omena, so that the shared attributes have more of its mass than that of omena.

$r(p(a), q(a))$	$p(a)$	$q(a)$	$f(a)$	$g(a)$	a
0.0188	0.0193	0.0182	16	3	`olla-subj-`, *be*
0.0155	0.0133	0.0182	11	3	`kuoria-obj-`, *peel*
0.0121	0.0121	0.0121	10	2	`syödä-obj-`, *eat*
0.0109	0.0060	0.0242	5	4	`ei-subj-`, *not*
0.0101	0.0085	0.0121	7	2	`-attr-iso`, *big*
0.0101	0.0085	0.0121	7	2	`kilo-mod-`, *kilo*
0.0085	0.0048	0.0182	4	3	`-cc-banaani`, *banana*
0.0083	0.0060	0.0121	5	2	`-attr-kuoria`, *peel*
0.0054	0.0048	0.0061	4	1	`-attr-pieni`, *small*
0.0046	0.0036	0.0061	3	1	`mansikka-cc-`, *strawberry*
0.0046	0.0036	0.0061	3	1	`ottaa-obj-`, *take*
0.0037	0.0024	0.0061	2	1	`ostaa-obj-`, *buy*
⋮	⋮	⋮	⋮	⋮	⋮
0.0024	0.0012	0.0061	1	1	`-cc-aprikoosi`, *apricot*
0.1645	0.1221	0.2848	101	47	

Table 3.19: The attributes `omena`, *apple*, shares with `appelsiini`, *orange*, ranked by their pointwise radius $r(p(a), q(a))$ for these words. Each a occurs $f(a)$ times with `omena` and $g(a)$ times with `appelsiini`, $p(a) = f(a)/827$ and $q(a) = g(a)/165$. The information radius $R(p, q)$ is $1 - 0.1645 = 0.8355$. The words share 30 attributes out of their 387 and 119 respective total attributes, with 827 and 165 occurences in all. For `omena` alone, see Table 3.11 on page 83. For `appelsiini`, see Table 3.12 on page 84.

$r(p(a), q(a))$	$p(a)$	$q(a)$	$f(a)$	$g(a)$	a
0.0237	0.0193	0.0296	16	41	`olla-subj-`, *be*
0.0151	0.0133	0.0173	11	24	`kuoria-obj-`, *peel*
0.0125	0.0121	0.0130	10	18	`syödä-obj-`, *eat*
0.0089	0.0085	0.0094	7	13	`kilo-mod-`, *kilo*
0.0082	0.0036	0.0289	3	40	`-attr-keittää`, *cook*
0.0077	0.0048	0.0137	4	19	`-cc-porkkana`, *carrot*
0.0074	0.0060	0.0094	5	13	`-attr-kuoria`, *peel*
0.0072	0.0072	0.0072	6	10	`myydä-obj-`, *sell*
0.0067	0.0157	0.0036	13	5	`-attr-kotimainen`
0.0059	0.0036	0.0108	3	15	`-cc-sipuli`, *onion*
0.0056	0.0048	0.0065	4	9	`saada-obj-`, *get*
0.0053	0.0085	0.0036	7	5	`-attr-iso`, *big*
⋮	⋮	⋮	⋮	⋮	
0.0009	0.0012	0.0007	1	1	`-attr-hauduttaa`, *stew*
0.2712	0.2830	0.3642	234	504	

Table 3.20: The attributes `omena`, *apple*, shares with `peruna`, *potato*, ranked by their pointwise radius $r(p(a), q(a))$ for these words. Each a occurs $f(a)$ times with `omena` and $g(a)$ times with `peruna`, $p(a) = f(a)/827$ and $q(a) = g(a)/1384$. The information radius $R(p, q)$ is $1 - 0.2712 = 0.7288$. The words share 105 attributes out of their 387 and 693 respective total attributes, with 827 and 1 384 occurrences. For `omena` alone, see Table 3.11 on page 83. For `peruna`, see Table 3.13 on page 85.

$r(p(a), q(a))$	$p(a)$	$q(a)$	$f(a)$	$g(a)$	a
0.1255	0.3083	0.0658	255	15	`-attr-vihreä`, *green*
0.0292	0.0193	0.0482	16	11	`olla-subj-`, *be*
0.0053	0.0024	0.0175	2	4	`olla-loc-`, *be*
0.0051	0.0060	0.0044	5	1	`ei-subj-`, *not*
0.0049	0.0024	0.0132	2	3	`tehdä-obj-`, *make*
0.0040	0.0036	0.0044	3	1	`ottaa-obj-`, *take*
0.0032	0.0012	0.0175	1	4	`tarvita-obj-`, *need*
0.0032	0.0012	0.0175	1	4	`-attr-vuosi`, *year*
0.0021	0.0012	0.0044	1	1	`antaa-subj-`, *give*
0.0021	0.0012	0.0044	1	1	`sanoa-obj-`, *say*
0.0021	0.0012	0.0044	1	1	`olla-comp-`, *be*
0.0021	0.0012	0.0044	1	1	`-attr-täydellinen`, *perfect*
0.0021	0.0012	0.0044	1	1	`osa-mod-`, *part*
0.0021	0.0012	0.0044	1	1	`sisältää-subj-`, *contain*
0.1931	0.3519	0.2149	291	49	

Table 3.21: The attributes `omena`, *apple*, shares with `vero#uudistus`, *tax reform*, ranked by their pointwise radius $r(p(a), q(a))$ for these words. Each a occurs $f(a)$ times with `omena` and $g(a)$ times with `vero#uudistus`, $p(a) = f(a)/827$ and $q(a) = g(a)/228$. The information radius $R(p, q)$ is $1 - 0.1931 = 0.8069$. The words share 14 attributes out of their 387 and 140 respective total attributes, with 827 and 228 occurrences. For `omena` alone, see Table 3.11 on page 83. For `vero#uudistus`, see Table 3.14 on page 86.

We saw that vihreä, *green*, is so important to omena, *apple*, for an unexpected reason: there is a puppet theatre group named Vihreä Omena, *Green Apple*. Many of the co-occurrences are on the culture pages.

A *green tax reform* may also sound unexpected, but the expression, vihreä verouudistus, does occur, along with *ecological tax reform*. Indeed, vihreä is the second most important attribute of verouudistus.

3.5 Ranking lists for the frequent nouns

Having computed the information radius of omena with appelsiini, with peruna, and with vero#uudistus, I can put the three words in the order of their decreasing similarity with omena. The result is a similarity ranking list:

omena, *apple*
0.729 peruna, *potato*
0.807 vero#uudistus, *tax reform*
0.835 appelsiini, *orange*

Here omena is said to be the *head word*, or just the *head*, of this list. The words ranked with respect to the head word are *tail words*, or just *tails*, of this list.

Neighbours of omena and appelsiini See Table 3.22 for the words nearest to omena, *apple* and appelsiini, with the criteria of nearness that I use.

The first thing to notice is that the neighbours are not too bad as for their similarity of meaning with the head word. Most of the neighbours are names of other kinds of fruit or vegetables, or at least food items. (Recall from Table 3.19 that both omena, *apple*, and appelsiini, *orange*, occur as objects of the verbs syödä, *eat*, and kuoria, *peel*.) The second thing to notice is that even the best similarity scores are not themselves very small: the nearest neighbour of omena, *apple*, is judged at 0.717, rather nearer to 1 than 0, and the nearest neighbour of appelsiini, *orange,* is judged to be only at 0.782. Scores near 0 are relatively rare.

Another distributional fact is that omena appears among the nearest neighbours of appelsiini, but appelsiini does not appear near omena. The similarity formula is indeed symmetric in its arguments, but there are many words that match omena better than any word matches appelsiini.

Ranking list of `omena`		Ranking list of `appelsiini`	
`omena`, *apple*		`appelsiini`, *orange*	
0.717	`tomaatti`, *tomato*	0.782	`sitruuna`, *lemon*
0.729	`peruna`, *potato*	0.787	`banaani`, *banana*
0.735	`paprika`, *bell pepper*	0.816	`tomaatti`, *tomato*
0.743	`papu`, *bean*	0.826	`peruna`, *potato*
0.769	`liha`, *meat*	0.835	`paprika`, *bell pepper*
0.791	`kala`, *fish*	0.835	`omena`, *apple*
0.794	`kasvis`, *vegetable*	0.836	`kanan#muna`, *(hen's) egg*
0.799	`ruoho`, *grass*	0.838	`sipuli`, *onion*
0.803	`banaani`, *banana*	0.839	`juusto`, *cheese*
0.807	`vero#uudistus`, *tax reform*	0.850	`liha`, *meat*
0.807	`marja`, *berry*	0.856	`kala`, *fish*
0.812	`lanka`, *thread*	0.857	`kesä#kurpitsa`, *aubergine*
0.813	`porkkana`, *carrot*	0.859	`muna`, *egg*
0.813	`kinkku`, *ham*	0.859	`kirjo#lohi`, *rainbow trout*
0.813	`muna`, *egg*	0.859	`kinkku`, *ham*
0.814	`vilja`, *grain*	0.862	`soija`, *soy*

Table 3.22: The nearest words to `omena`, *apple*, in the order of their increasing information radius with the head word.

Tax reform Another noticable point in Table 3.22 is that `omena`, *apple*, has two strikingly odd neighbours. The first is `verouudistus` *tax reform*, ranked tenth, discussed above.

Thread The second, less striking but still unexpected, is `lanka`, *thread*, which is ranked twelfth. The reason for its similarity to `omena`, *apple*, again appears to be the importance of the attribute `-attr-vihreä`, *green*, to both: there is a publication with the name `Vihreä Lanka`.

Chapter 4

Identifying semantically similar words using the information already at hand

Many head words in the similarity table rank both good and bad tail words among their most similar. By a *good tail* I mean a word that is *semantically similar* to the head of the list; a *bad tail*, then, is *not* semantically similar to the head. It would be an improvement if some of the bad tails could be filtered out of the table without also losing many good tails. A more cautious procedure would be to simply identify the best and the worst.

The similarity ranking table does not encode all the information that is available about a pair of words; it only contains the similarity scores and ranks. Nevertheless, there are the individual representations of the words, and their overlap is readily computed. In this chapter, I explore the possibility of using some of that additional information to discriminate between the good pairs and the bad.

Section 4.1 reviews the similarity table through a random sample of 30 relatively frequent nouns, though not among the most frequent, each taken both as the head word of its own list and as a tail word on those lists where it occurs near the top.

Section 4.2 presents a simple random sample, taken from the full similarity table, of 400 head–tail pairs where the tail word is among the 20 most similar to the head word. These are *distributionally* similar pairs, and the rest of the chapter aims to identify the semantically good and bad pairs among these.

Section 4.3 gives my intuitive classification of each of these 400 random pairs as semantically `good` or `bad`, and also as `sure` or `unsure` to indicate whether I felt sure about the primary classification. The semantic class of a pair becomes the output variable of `Sense`, whose value is to be predicted.

Moreover, my certainty is the auxiliary variable of `Ease` that can be used to exclude difficult pairs.

The classification of all 400 pairs is recorded in Appendix B. A subsample of 16 pairs, four from each class, is used as examples throughout this chapter. Other words used are familiar by now: `omena`, *apple*, `appelsiini`, *orange*, `peruna`, *potato*, and `vero#uudistus`, *tax reform*.

In Section 4.4, I identify a number of distributional input variables used to predict the semantic class of `Sense`. These include the information radius which I use as a similarity score, now labeled `Sim`, and the ranks of `Head` and `Tail` with respect to each other, labeled `Rank` and `Knar`. More important in this chapter are the number of attributes that occurred with each word, labeled `NHead` and `NTail`, the number of shared attributes, labeled `NShared`, and the proportion of its probability mass that each word assigns to the shared attributes, labeled `PHead` and `PTail`.

Section 4.5 explores the distribution of the different input variables for the two values, `good` and `bad`, of the output variable `Sense`. A simple visual comparison of their estimated density curves suggests that several input variables display a modest difference in the expected direction, but the overlap of the curves is also large.

Section 4.6 explores a number of tree models, trained on the classified sample of 400 pairs. These models attempt to predict the semantic class `Sense` from the values of some of the input variables. I experiment with different combinations of input variables. I also try training the models with only the pairs that I found easy to classify. The models are displayed graphically in this chapter and in a textual form (provided by the `rpart` library of R) in Appendix C.

Each model is evaluated anecdotally by observing its classification of four tails of `omena`, *apple*, and more systematically by five different success rates in predicting the classes given in the training data. The performance of the models on easily classified pairs is recorded separately.

Finally, in Section 4.7, a separate sample of 100 more pairs is set up and classified, the previously trained models are run against this test data, and two 'best' models are identified: one of the models seems to be the best in three of the prediction tasks, another in the remaining two. The 100 test pairs with their four semantic classes are also recorded in Appendix B, after the 400 training pairs.

4.1 A semantic assesment of a sample

There is now a ranking list of the one hundred *distributionally* most similar words for each of my words. This section is an attempt at a rough qualitative evaluation of the *semantic* similarity of some of the best head–tail pairs. I took a random sample of 30 words and collected all the pairs where (1) the head word was in the sample and the tail word was in the top three of that list, or (2) a word in the sample was one of the top three tail words.

The 30 frequent words are a random sample from the ninth frequency decile range of our vocabulary of the nouns that occurred more than one hundred times, together with the few words below and above that are tied at their respective deciles. This means that a little less than 10% of our frequent nouns were too frequent to be eligible (were above the ninth decile range, occurred more than 1 330 times), and a little less than 80% were not frequent enough (were below the ninth decile range, occurred less than 651 times).

The eligible set consisted of 1 803 nouns, and the random sample itself consists of the initial 30 words in a random shuffle of the eligible set. They are listed in Table 4.1, together with the number of head words that have them among their three first tail words.

The sample turns out to contain one of the words that had to be excluded for technical reasons: `tarpeen`. It does not have a ranking list, nor does it occur in the ranking list of any other word, so there is not much to say about it. It is described below briefly.

The bulk of the remaining 29 nouns consists of 15 common nouns and 13 proper nouns. The 13 proper nouns are further divided into 5 names of geographical locations, 7 first or last names of a person, and 1 name of a company. The one remaining word, `Tieto`, turns out to be problematic for this combination of corpus and parser.

In the tables of this section, I use the following three symbols to express my intuition about the semantic nature of the distributional head–tail pairs:

1. a smile ⌣ for semantic similarity;

2. a frown ⌢ for no semantic similarity;

3. a stymied expression ≍ for those that I find difficult to judge.

These appear mainly in Tables 4.2 and 4.3 for the common nouns in the sample, and in Table 4.8 on page 116 that summarises the results. In the summary, a word can have any combination of these three symbols, because I judge at least three different tails for each, and possibly several heads, as

The number of heads that rank the word among three	The random word of high frequency (651–1 330 occurrences in the corpus)
5	`lento#yhtiö`, *airline*
8	`pahoin#pitely`, *physical abuse*
1	`kesto`, *duration*
1	`side`, *bandage, bond*
1	`markkina#korko`, *market rate of interest*
1	`todistaja`, *witness*
7	`tasku`, *pocket*
6	`vauva`, *baby*
11	`vuokralainen`, *tenant*
1	`ihmiskunta`, *humankind*
2	`maa#pallo`, *globe*
1	`tango`, *tango*
1	`tuntemus`, *knowledge, feeling*
8	`tulva`, *flood*
3	`konservatiivi`, *conservative*
3	`vihd`, *Vihti*
6	`Boston`, *Boston*
14	`Kalifornia`, *California*
5	`Manchester`, *Manchester*
1	`Marjaniemi`, *Marjaniemi*
7	`Joni`
0	`Jim`
10	`Bildt`
2	`Kononen`
7	`Martikainen`
3	`Antonio`
2	`Saku`
6	`Rautakirja`
1	`Tieto`
–	`tarpeen`

Table 4.1: A random sample of frequent nouns, in five groups in their order in a random shuffle, from roughly the 9th frequency decile range of those nouns that occurred over a 100 times in the corpus (frequency range 651–1 330) with the number of head words that have them among the three first tails.

well. The result is that all but one of the words have the smile, while only a few have the frown.

The excluded noun One of the words in the sample, `tarpeen`, is bad. It is one of the words that have no attributes, and therefore could not be represented as a probability distribution. Second, it is not a good noun: it seems to occur only in the expression `olla tarpeen`, *be needed*, and it is not clear whether the parser is meant to analyse it as the genitive of the noun `tarve`, *need*, or as an adverb. (Both analyses occur elsewhere in the corpus. The former go into the representation of `tarve`; the latter do not go into the representation of any noun.)

The fifteen or so common nouns in the sample Table 4.2 lists the first three tail words in the distributional ranking lists of the common nouns in the random sample. Attached to each tail word is a symbol indicating my satisfaction concerning the semantic similarity of the head and the tail.

The following are a few comments on these judgments:

1. Most of the tails are simply good. Even if they are not useful for paraphrasing the head word, they mean the right *kinds* of thing.

2. In some of the cases where I hesitate, the only problem is that the tail word is a much more generic than the head word. This is the case especially with `henkilö`, *person*, being similar to `todistaja`, *witness*. A witness is such a specific kind of person that the relation feels too weak.

3. A few of the words are clearly ambiguous about their meaning. As long as there is an appropriate relation between the appropriate senses, I accept the pair. The ambiguity still warrants attention. Such words are at least `yritys`, *company, attempt*, `korko`, *interest (on money), heel (of shoe)*, `tuntemus`, and `kokemus`, possibly also `maa`, *country, land, earth* and `suhde`, *relation or proportion*, respectively. Of these, `tuntemus`, occurs here as the head word. (In Finnish, `pankki`, *bank* and `vauva`, *baby, small child*, are not ambiguous the way they are in English. The latter does occur in cross-lingual jokes where, for example, *my baby was gone* is translated as if it had been *my child is a gun*, but then the point is precisely that the Finnish `vauva` can not refer to an adult woman, however dear to the singer.)

4. One dubious neighbour, `tilanne`, *situation*, occurs twice as a tail: with `side` and with `tulva`, *flood*.

The head (in the sample)	—	The first three tails of the head
ihmiskunta, *mankind*	—	◡kansa#kunta, *nation*, ◡kansa, *people*, ◡yhteis#kunta, *society*
kesto, *duration*	—	◡pituus, *duration*, ◡odotus#aika, *waiting time*, ◡käsittely#aika, *handling time*
konservatiivi, *conservative*	—	◡sosiaali#demokraatti, *social democrat*, ◡sosialisti, *socialist*, ◡demari, *democrat*
lento#yhtiö, *airline company*	—	◡pankki, *bank*, ◡yritys, *company or attempt*, ◡yhtiö, *company*
maa#pallo, *globe*	—	◡maailma, *world*, ◡maailman#kaikkeus, *universe*, ◡maa, *earth*
markkina#korko, *market interest*	—	◡korko, *interest or heel*, ◡helibor-#korko, ⌒oppi#määrä, *syllabus*
pahoin#pitely	—	◡petos, *fraud*, ◡vahingon#teko, ◡varkaus, *theft*
side	—	≈kontakti, *contact*, ◡suhde, *relation*, ⌒tilanne, *situation*
tango, *tango*	—	◡musiikki, *music*, ◡iskelmä, ◡kansan#musiikki
tasku, *pocket*	—	◡pussi, *bag*, ≈käsi, *hand*, ◡piilo
todistaja, *witness*	—	◡silmin#näkijä, *eye witness*, ◡asian#tuntija, *expert*, ≈henkilö, *person*
tulva, *flood*	—	≈tilanne, *situation*, ◡järistys, *quake*, ≈lama, *depression*
tuntemus	—	◡kokemus, ◡tunne, *feeling, sensation*, ◡asian#tuntemus, *expertese*
vauva, *baby, small child*	—	◡lapsi, *child*, ◡potilas, *patient*, ≈ihminen, *human*
vuokralainen, *tenant*	—	◡asiakas, *customer*, ◡ostaja, *buyer*, ◡osakas

Table 4.2: The common nouns in the sample (left column) and their first three tails. The symbols attached to the tails indicate my satisfaction of the semantic similarity of the head and the tail.

Table 4.3 lists the head words that have the 15 common nouns in the sample among the first three of their tail words. Attached to each head word is a symbol indicating my satisfaction with the semantic similarity of the head and the tail.

The following are a few comments on these judgments:

1. The number of heads that have the sampled words in their first three tails ranges from the 0 of `Jim` to the 14 of `Kalifornia`. This is not unexpected.

2. Some of these heads also appeared in the first three tails of the sampled words. Some are new.

3. Most of these heads are again easy to accept as semantically more or less similar to their sampled tail.

4. Three unexpected words turn up: `varoitus#aika`, *warning time*, for `markkina#korko`, *market interest*; `Vale`, *???* for `side`, *???*; `mono`, *ski boot?*, for `vuokralainen`, *tenant*. One of these, `side`, *bond*, had already presented problems.

5. Should I accept `vasikka`, *calf*, for `vauva`, *baby, small child*? Neither decision feels right.

The meaning ambiguities among the fifteen common nouns One ambiguous word in our sample is `tuntemus`, which can refer to a knowledge of some field, or to a feeling or sensation. Others that turn up through similarity are `yritys`, *company or attempt*, and `korko`, *interest or heel*.

At least we can happily accept `kokemus`, *experience*, `tunne`, *feeling*, and `tunto`, *sense of touch*, as similar to `tuntemus`. Maybe we could accept them as similar to each other, though not as satisfactorily. However, `asian#tuntemus`, *expertice*, would not match `tunne` or `tunto` well, I think.

It appears that `yritys` attracts almost only words that are more like *company* than *attempt*. Closest to the *attempt* sense might be `hanke` and `työ`. Occurrences of `korko` as *heel* are rare compared to its occurrences as *interest*.

The semantic failures of the fifteen common nouns The compound word `markkina#korko`, *market interest*, ranks as the third highest `oppi#määrä`, *syllabus??*, and is ranked the third highest by `varoitus#aika`, *warning time??*. Both judgments are based on only six shared attributes, with *short* being the strongest among them. The other shared attributes for *warning*

The heads that have the tail in their first three	—	The tail (in the sample)
⌣kansa#kunta, *nation*	—	ihmiskunta, *mankind*
⌣pituus, *length*	—	kesto, *duration*
konservatiivi#hallitus, *conservative government*, ⌣nationalisti, *nationalist*, ⌣sosialisti, *socialist*	—	konservatiivi, *conservative*
investointi#pankki, *investment bank*, öljy-#yhtiö, *oil company*, raha#laitos, *monetary institution*, tele#visio#yhtiö, *television company*, tv-#yhtiö, *tv company*	—	lento#yhtiö, *airline company*
⌣Jupiter, *Jupiter*, ⌣maailman#kaikkeus, *universe*	—	maa#pallo, *globe*
⌢varoitus#aika,	—	markkina#korko, *market interest*
⌣kidutus, *torture*, ⌣kiristys, *blackmail*, ⌣murha, *murder*, ⌣raiskaus, *rape*, ⌣ryöstö, *robbery*, ⌣tappo, *killing*, ⌣vahingon#teko, ⌣varkaus, *theft*	—	pahoin#pitely, *physical abuse*
⌢Vale	—	side
⌣valssi, *waltz*	—	tango, *tango*
⌢hiha, *sleeve*, ⌢kainalo, *armpit*, kätkö, ⌣kukkaro, lokero, lompakko, *wallet*, pussi, *bag*	—	tasku, *pocket*
⌣silmin#näkijä, *eye witness*	—	todistaja, *witness*
⌣kuivuus, *draught*, ⌣lumi#myrsky, *snowstorm*, ⌣metsä#palo, *forest fire*, ⌣myrsky, *storm*, ⌣nälän#hätä, *famine*, ⌣pyörre#myrsky, *hurricane*, ⌣rankka#sade, ⌣vyöry	—	tulva, *flood*
⌣tunto	—	tuntemus
⌣esikoinen, *first-born*, ⌣pentu, *cub*, ⌣pienokainen, ⌣pikku#lapsi, *small child*, ⌣sikiö, *foetus*, ⌢vasikka, *calf*	—	vauva, *baby, small child*
apteekkari, *apothecary*, haltija, *possessor*, isännöitsijä, joukkue#toveri, *team-mate*, kanta-#asiakas, luovuttaja, *donor or quitter*, ⌢mono. ⌢puutarhuri, *gardener*, takaaja, *backer*, ⌣vuokraaja, vuokran#antaja	—	vuokralainen, *tenant*

Table 4.3: The common nouns in the sample (right column) and the words that have them in their first three tails. The symbols attached to the heads indicate the semantic similarity of the head and the tail.

time are *be*, *month*, *become*, *accept*, and *moment*; for *syllabus*: *be*, *long*, *be*, *not*, and *correspond*. The basis for similarity seems to be only that the two concepts have some kind of length.

The singularly odd match between `Vale` and `side` is based on a single shared attribute, `-attr-port`. The relevant occurrences of `side` are not in the sense of *bond* or *bandage* at all. Instead, they refer to a restaurant called `Port Side`, where bands perform music at ten o'clock. (I was luckier with `Saku` and `Susa`. They, too, share only one attribute.)

The word `tilanne`, *situation*, has 5 272 attributes, a rather large number. (It was not in our sample, so it need not be in that frequency band.) It shares 141 of them with `side`, *bond*, which has 425 attributes. Apart from the different dependency relations with *be* and *not*, the most important shared attributes include *economic*, *political*, *new*, and *current*. These seem to be acceptable, even though the resulting high similarity is questionable.

Contrasting this with the word `suhde`, *relation*, which has 5 273 attributes and shares 147 of them with `side`, *bond*, results in much the same numbers as those of `tilanne`, *situation*, yet the words seem a much better match in terms of their meanings. Again apart from *be* and *not*, the most important shared attributes include `välinen`, *between??*, `läheinen`, *close*, `sosiaalinen`, *social* and `keskinäinen`, *mutual*. These seem indicative of the meanings of the two words. (The dependency relation label in each case is `attr`.)

(In another contrast, the match between `tulva`, *flood*, and `tilanne`, *situation*, is appropriate, though one of the words is much more specific than the other.)

Why would `mono`, *ski boot*, rank `vuokralainen`, *tenant*, so high? Its two highest words are `suksi`, *ski*, and `kenkä`, *shoe*, and *tenant* is third. There is no obvious explanation for this. Possibly, it is the general nature of the shared attributes, which include *new*, *old*, *find*, and *choice*, together with the flat frequency distribution of `mono`: it has 105 attributes from 123 occurrences, and its most frequent attributes occur with it only three times.

The five place names in the sample Geographical names are mildly interesting. The words that they rank are in Table 4.4, and the words that rank them, in Table 4.5. It seems the names of big cities will often be similar to names of big cities, which I find to be more or less acceptable.

A closer look reveals that these names often share some strong attributes that occur repeatedly with these particular names. For example, `Boston` shares *symphony orchestra* and *university* with `Toronto`, and *university* and *marathon* with `Chicago`. Another word with *university* as a strong attribute is `Kalifornia`, *California*, which makes it similar to `Tartto` in particular; it

The head (in the sample)	—	The first three tails of the head
vihd, *Vihti*	—	Sipoo, Tuusula, Lohja
Boston	—	Toronto, Chicago, Detroit
Kalifornia, *California*	—	Michigan, Tartto, Florida
Manchester	—	Dundee, Sheffield, simmis, *???*
Marjaniemi	—	uima#stadion, *swimming stadium*, ydin#keskusta, *nuclear? centre*, messu#keskus, *exhibition centre?*

Table 4.4: The words that the place names in the sample rank among three.

The heads that have the tail in their first three	—	The tail (in the sample)
Halikko, Lempäälä, Valtimo	—	vihd, *Vihti*
Charlotte, Chicago, Denver, Miami, portland, Toronto	—	Boston
Alaska, Arizona, Edinburgh, Florida, Massachusetts, Minnesota, Ohio, Oklahoma, Osaka, Teksas, *Texas*, Tennessee, Texas, Utah, Virginia	—	Kalifornia, *California*
Bristol, Dundee, manch, Sheffield, simmis, *???*	—	Manchester
Isokari	—	Marjaniemi

Table 4.5: The words that rank the place names in the sample among three.

is also an osa#valtio, *state*, and therefore similar to other states.

The Finnish municipalities Sipoo, Tuusula and Lohja all share with vihd, *Vihti*, the word kunta, *municipality*, as a very strong attribute, together with other words referring to local government. The parser failed to find the right capitalised base form Vihti, but it found appropriate dependency relations, so the incorrect base form does not pose a problem.

In addition to other names, geographical names attract descriptive words that refer to places. For example, Lempäälä ranks onnettomuus#paikka, *scene of accident*, as its third closest word. (It ranks vihd, *Vihti*, in second place.) One of our sampled names, Marjaniemi, refers to a part of Helsinki and ranks *swimming stadium*, *nuclear centre* and *exhibition centre*, with the name Helsinki as the strongest shared attribute with them all.

The word simmis is a name of a swimming team, and the base form should really retain the capitalization of the surface form, Simmis. What also occur are: Simmis United and Simmis U, and it is these two attributes, United and U that Simmis shares, strongly, with Manchester. There are seven shared attributes, but those two dominate the similarity. One of the others is the verb *to win*. Perhaps we need to accept that Manchester is ambiguous.

Many of the attributes of simmis are the names of the swimmers, not from ordinary sentences, but from tabular material where the parser has linked the name of the team to the name of the player, as is indicated below:

```
22  Jenni       Jenni       attr:>23  &A> N SG NOM
23  Koivuniemi  Koivuniemi  attr:>24  &A> N SG NOM
24  Simmis      simmis                &NH N SG NOM
25  9.51,16     9.51,16     mod:>24   &NH NUM CARD
```

The seven person names in the sample The names of people, first or last, in the sample attract acceptable company in a rather unexciting way. First names become similar on the strength of shared last names, and last names on the strength of shared first names.

The sampled names rank among the highest three the words in Table 4.6; the words in Table 4.7 rank the sampled names among the highest three.

The one company name in the sample The company name Rautakirja ranks other company names Panostaja, Instrumentarium, and Finvest. The shared attributes are essentially A and B, which I guess have something to do with the stock market.

The company name Rautakirja is ranked by Finvest, Instrumentarium, International, Panostaja, Stockmann, and WSOY, which are all company

The head (in the sample)	—	The first three tails of the head
Joni	—	Raine, Taneli, Atso
Jim	—	James, Ian, Melissa
Bildt	—	Lewis, Öhman, Lipponen
Kononen	—	Essayah, Sievinen, Suomalainen
Martikainen	—	Salmi, Rissanen, Kiuru
Antonio	—	Manuel, Jesus, Carlos
Saku	—	Rudolf, Susa, *Susan?*, canadiens???

Table 4.6: The seven person names in the sample, and their first three tails. First names are similar to first names, and last names to last names.

The heads that have the tail in their first three	—	The tail (in the sample)
Atso, Elina, Kim, Marjut, Raine, Riku, Taneli	—	Joni
broek, *???*, Gaidar, Gustaf, Hamilton, Kinnock, Lewis, Ludvig, Mason, Öhman, Stoltenberg	—	Bildt
Lindman, Vaala	—	Kononen
Jalonen, Karhunen, Lamberg, Lehtola, Ovaska, Pietarinen, Tikka	—	Martikainen
Diego, Jesus, Saara	—	Antonio
Rudolf, Susa, *Susan?*	—	Saku

Table 4.7: The sampled person names (right column) and the heads that have them in their first three tails. Given names are similar to given names, and family names to family names.

names.

The one really failing word in the sample The most spectacular oddity of the whole sample is Tieto. It ranks jolla, *jolly, yawl*, and Rata, and tää, *this*. See below for additional information on jolla and tää. In the other direction, Tieto is ranked by Ulmanen.

Often Tieto occurs as a common noun, meaning *knowledge* or *information* or *datum*, and its capital first letter is due to a sentence-initial position. It is not at all clear to me why the parser reduces some of these to tieto and some to Tieto. Another frequent type of occurrence, however, is as a part of a newspaper department name Tieto & kone, an apparent play on tietokone, *computer*, analysed so that kone becomes an attribute. (There are other repeated titles that also begin with Tieto.) Then there is a company called TT Tieto Oy, analysed so that oy becomes an attribute.

The real noun jolla, *jolly*, is not frequent in the corpus. Many of the occurrences are the relative pronoun joka, *that*, in the adessive case, not a noun. It is important to remember that the parser was an early version.

Similarly, Rata occurs as a common noun, meaning *railway* or any of those courses where horses or cars or other sprinters race, capitalised at the beginning of a sentence, and should have been reduced to rata in such occurrences. It is also short for Rahoitustarkastus, and in those occurrences, it is properly capitalised. (It also occurs in the name of a horse, Rata Rosvo.)

The tää, *this*, is a spoken form of the pronoun tämä, *this*, but occurs sufficiently often in the newspaper. The parser did not expect it and – reasonably but incorrectly – guessed it to be a noun, as found below:

```
2  Tää         tää         subj:>3  &NH N SG NOM
3  onkii       onkia       main:>0  &+MV V ACT IND PRES SG3
4  puhistamo   puhistamo            &NH N SG NOM
```

And Ulmanen is the last name of several people, but also occurs in Ulmanen & Roiha. These occurrences of & have been encoded in a part of the corpus as &, the parser has made Amp an attribute of Ulmanen, and the same happened to Tieto. The words share only three attributes, of which Amp alone contributes over 75% of the information radius.

Summing the assessment up Table 4.8 summarises this evaluation of the semantic nature of our distributional ranking table. Overall, the pairs are easy to accept. Note that at least one word in these pairs was very frequent, and that these are the highest ranking pairs in their lists.

Satisfaction			The words in the sample
☺			`lento#yhtiö`, *airline*
☺			`pahoin#pitely`, *physical abuse*
☺			`kesto`, *duration*
(☺)	😟	😕	`side`, *bandage or bond*
☺	😟		`markkina#korko`, *market rate of interest*
☺			`todistaja`, *witness*
☺		(😕)	`tasku`, *pocket*
☺			`vauva`, *baby*
☺	(😟)	(😕)	`vuokralainen`, *tenant*
☺			`ihmiskunta`, *humankind*
☺			`maa#pallo`, *globe*
☺			`tango`, *tango*
☺			`tuntemus`, *knowledge or feeling*
☺			`tulva`, *flood*
☺			`konservatiivi`, *conservative*
☺			`vihd`, *Vihti*
☺			`Boston`, *Boston*
☺			`Kalifornia`, *California*
☺		😕	`Manchester`, *Manchester*
(☺)		😕	`Marjaniemi`, *Marjaniemi*
☺			`Joni`
☺			`Jim`
☺			`Bildt`
☺			`Kononen`
☺			`Martikainen`
☺			`Antonio`
☺	😟		`Saku`
☺			`Rautakirja`
		😕	`Tieto`
			`tarpeen`

Table 4.8: This is my overall evaluation of the sampled words: a summary of the previous tables. Almost all sampled words have several good similar words, and only a few have clearly bad words. Parenthesised smilies appear in the data but do not have a significant effect on my overall satisfaction.

4.2 A training sample of four hundred pairs

I took a simple random sample of four hundred pairs of relatively high *distributional* similarity. These should not be too many to classify by hand as intuitively good or bad, in the sense of *semantic* similarity. The classification is described below in Section 4.3. This sample is independent of the illustrative sample of the previous section. In fact, I made it much earlier in time, and I classified it before studying the underlying data. The later testing sample of an additional one hundred pairs is also independent.

I adopt the formal variable name `Head` for the head word and the name `Tail` for tail word. These identify a row in a data frame:

`Head`	the head of a ranking list in our table; one of the 17 835 nouns that I have, but excluding names.
`Tail`	a tail in a position from 1 to 20 in that list; one of the 16 857 nouns that occur in those positions of the table, but excluding names.

Names were excluded by removing from the list of eligible head–tail pairs those that contained at least one capital letter.

Random sampling, as usual, is expected to produce a data set that is statistitically representative of the larger population.

A sub-sample of sixteen pairs A sub-sample of 16 pairs from the sample of 400 serve as examples of the classification throughout the chapter. These represent the four different outcomes of the classification that was conducted by hand.

Focus on precision The full similarity table contains the ranking lists of 100 tail words each. I restricted the sampling to the first 20 tail words in each list, leaving me less than 400 000 head–tail pairs, and took a sample of 400 from these.

This can be seen as a focus on *precision*: the proportion of semantically similar pairs among those that are distributionally highly similar. Very little can be said about the complementary concept of *recall*: the proportion of the good pairs in our top-twenty lists among the pairs that I would classify as good.

Vocabulary size Most of the vocabulary can occur as tail words in the sample. The lists of 100 tail words contain 17 707 different words out of the 17 835 possible. The lists of 20 tail words contain 16 857 different words.

Defective lists A handful of heads fail to have many interesting tails. We have 67 lists of 100 tail words that end with the similarity score of 1.0, which indicates no similarity at all. Of these 67 lists, as many as 46 reach 1.0 already in their prefix of 20 tails.

4.3 The semantic output variable

Appendix B, page 175, lists my intuitive judgments of semantic closeness for all 400 pairs in the sample in four tables. Some pairs were difficult, so I classified them in two ways: into 'good' or 'bad' cases, and into 'sure' and 'unsure' cases.

`Sense`	my intuitive judgement of the semantic similarity of `Head` and `Tail`, in {`bad`, `good`}.
`Ease`	my intuitive judgement of how easy it was for me to decide on the value of `Sense`, in {`sure`, `unsure`}.

The 400 pairs are distributed as follows:

	`sure`	`unsure`	$\sum$
`good`	140	88	228
`bad`	125	47	172
$\sum$	265	135	400

Many pairs were easy to decide. If a pair was clearly semantically similar, I labeled it as `good` for `Sense` and as `sure` for `Ease`. If it was clearly not semantically similar, I labeled it as `bad` for `Sense` and, again, as `sure` for `Ease`.

Many pairs were not easy to decide. These I labeled as being `unsure` for `Ease`.

The whole grade of a pair consists of two variables, `Sense` with the values of `good` and `bad`, and `Ease` with the values of `sure` and `unsure`. Furthermore, I could move some pairs to another class without any conviction of having made the classification better or worse.

The most interesting cases should be those that I was sure about. In addition, it would be useful to find a method of removing the bad cases while keeping the good cases. The tree models that I build and evaluate in Sections 4.6 and 4.7 are my attempt to do that.

The sub-sample of example pairs Table 4.9 gives four randomly chosen example pairs of each class. Table 4.10 shows my classification of these pairs again, approximately in a format that can be read into R as a data frame.

Table 4.9: My intuitive cross-classification of the sub-sample of 16 from the full sample of 400 pairs, with `Head` on the left, and `Tail` on the right. The headings give the semantic class as 'good' or "bad" and the ease of classification as 'sure' or 'unsure'. Table 4.10 shows the classification in the form of a data frame.

Good/sure (4 of the 140 in Table B.1, page 176)

leivonnainen	kakku
päättäjä	luottamus#henkilö
oikeus#käytäntö	laki
strategia	ohjelma

Good/unsure (4 of the 88 in Table B.2, page 180)

isku#ryhmä	jury
velka#kirja	omaisuus
yhtiö#kokous	istunto
palvelu#työn#antaja	työn#antaja#liitto

Bad/unsure (4 of the 47 in Table B.3, page 183)

komeus	ikä
asunto#tuotanto	väki#luku
perus#rakenne	perus#asia
storgårds	ekki

Bad/sure (4 of the 125 in Table B.4, page 185)

vasta#kaiku	super#bingo
sankar	asia
ramppi	asunto
hitunen	disk#ontto#korko

Table 4.10: My intuitive cross-classification of the sub-sample of 16 pairs in the form corresponding to a data frame.

Head	Tail	Sense	Ease
leivonnainen	kakku	good	sure
päättäjä	luottamus#henkilö	good	sure
oikeus#käytäntö	laki	good	sure
strategia	ohjelma	good	sure
isku#ryhmä	jury	good	unsure
velka#kirja	omaisuus	good	unsure
yhtiö#kokous	istunto	good	unsure
palvelu#työn#antaja	työn#antaja#liitto	good	unsure
komeus	ikä	bad	unsure
asunto#tuotanto	väki#luku	bad	unsure
perus#rakenne	perus#asia	bad	unsure
storgårds	ekki	bad	unsure
vasta#kaiku	super#bingo	bad	sure
sankar	asia	bad	sure
ramppi	asunto	bad	sure
hitunen	disk#ontto#korko	bad	sure

4.4 Distributional input variables

This section involves the building of an inventory of numerical variables that reflect the distributional characteristics of pairs of words. Each pair comes from a ranking list and consists of the head word of the list and some tail word.

The next step in this process is to establish variables that can be read off the ranking table itself: the information radius and the ranks of the head and tail of the pair with respect to each other.

After that, I establish variables that might provide more information: the number of attributes that the words in the pair have or share and the proportions of their probability masses that belong to those shared attributes.

From the ranking table The full ranking table for my vocabulary would list all words as tails for all words, in the order of decreasing similarity. I have all words as heads, but the initial lists of the 100 tails only.

The ranking table contains the information radius for each head–tail pair in it. Additionally, the rank of the tail with respect to a head is simply its position on the list. The exact rank of a head with respect to a tail is only available if it is at most one hundred. I adopt the following three as variables:

`Sim`	the information radius of `Head` and `Tail`, in $[0\,..\,1]$; smaller values indicate a higher distributional similarity.
`Rank`	the position of `Tail` on the ranking list of `Head`, in $\{1, 2, 3, \ldots\ 19, 20\}$.
`Knar`	the position of `Head` on the ranking list of `Tail`, in $\{1, 2, 3, \ldots, 99, 100, >100\}$.

The unknown ranks, reported as >100, had to be encoded as numbers in the data frame. I chose 1 000 000, though of course I know that the true rank would be at most the number of words.

It is important to note that `Rank` is small for all of these pairs. A large `Knar`, however, may reveal something interesting about the pair: the association between the words seems to be asymmetric. The symmetry in question has been used by others to identify particularly good pairs, so its absence *might* suggest badness.

Numbers of attributes Let us now turn to the additional information that is not available in the ranking table, but is still present in our computational representations of the words in it.

First, the attributes can be merely counted, more or less ignoring their weights. Three different numbers of attributes might affect confidence in a similarity judgment:

`NShared`	the number of the attributes that `Head` and `Tail` share, between 0 (inclusive) and min(`NHead`, `NTail`) (inclusive)
`NHead`	the number of `Head` attributes, in $\{1, 2, \dots\}$
`NTail`	the number of `Tail` attributes, in $\{1, 2, \dots\}$

I might be suspicious of a high relative similarity computed from a small number of shared attributes. The ratios of `NShared` to `NHead` and `NTail` might also reveal something; I will try them briefly in Section 4.6.3, where they are called `HShared` and `TShared`.

At this time, let us proceed to identify two variables that take the weights into account.

Shared proportions of weight The total weight of a word is not interesting: I have normalised them so that the sum is always 1.0. The total weights of the *shared* variables, separately for `Head` and `Tail`, are interesting:

`PHead`	the proportion of its weight that `Head` shares with `Tail`, in $[0\,..\,1]$
`PTail`	the proportion of its weight that `Tail` shares with `Head`, in $[0\,..\,1]$

The attributes here have also a kind of 'shared weight', which I have called the 'pointwise radius'. Their sum is just $1 - R$, which would be redundant with the information radius R itself as a variable, `Sim`.

Some familiar examples Now I have, for each pair of `Head` and `Tail`, eight different numbers that somehow characterise their similarity. Table 4.11 shows these variables for the head word `omena`, *apple*, and the four tail words with it that were discussed earlier: `appelsiini`, *orange*, `lanka`, *thread*, `peruna`, *potato*, and `vero#uudistus`, *tax reform*, respectively.

The two rank variables, `Rank` and `Knar`, are available in the ranking lists. Three of these in Table 4.11 can be seen on the two lists in Table 3.22 on page 100: `peruna` and `vero#uudistus` are seen on the list for `omena` there,

Head	Tail	Rank	Knar	Sim
omena	appelsiini	36	6	0.8355
omena	lanka	12	>100	0.8118
omena	peruna	2	16	0.7288
omena	vero#uudistus	10	>100	0.8069

Head	Tail	NShared	NHead	NTail	PHead	PTail
omena	appelsiini	30	387	119	0.1221	0.2848
omena	lanka	24	387	374	0.3821	0.3673
omena	peruna	105	387	693	0.2830	0.3642
omena	vero#uudistus	14	387	140	0.3519	0.2149

Table 4.11: The eight variables and their values for the head word `omena`, *apple*, and the four tail words `appelsiini`, *orange*, `lanka`, *thread*, `peruna`, *potato*, and `vero#uudistus`, *tax reform*, respectively.

and `omena` is seen on the list for `appelsiini`. The corresponding values of `Sim` can also be seen there, or alternatively computed from the shared attribute weights.

Three of the variables are essentially the column sums in a table that lists the weights of shared variables: `Sim` is one minus the sum of shared weights, `PHead` is the sum of head weights and `PTail` the sum of tail weights. A fourth variable, `NShared`, is the number of rows in such a table.

Table 4.12 shows this for the head `omena`, *apple*, and its bad tail `vero#uudistus`, *tax reform*. Table 4.13 shows it for the good tail `peruna`. The captions of these tables also give the remaining two variables `NHead` and `NTail` that are obtained, trivially, from the separate representations of `Head` and `Tail`.

Shared	**Head**	**Tail**	**Attribute**
0.1255	0.3083	0.0658	`-attr-vihreä`
0.0292	0.0193	0.0482	`olla-subj-`
0.0053	0.0024	0.0175	`olla-loc-`
0.0051	0.0060	0.0044	`ei-subj-`
0.0049	0.0024	0.0132	`tehdä-obj-`
0.0040	0.0036	0.0044	`ottaa-obj-`
0.0032	0.0012	0.0175	`tarvita-obj-`
0.0032	0.0012	0.0175	`-attr-vuosi`
0.0021	0.0012	0.0044	`sisältää-subj-`
0.0021	0.0012	0.0044	`sanoa-obj-`
0.0021	0.0012	0.0044	`osa-mod-`
0.0021	0.0012	0.0044	`olla-comp-`
0.0021	0.0012	0.0044	`antaa-subj-`
0.0021	0.0012	0.0044	`-attr-täydellinen`
0.1931	0.3519	0.2149	

Table 4.12: The weights of the 14 shared attributes of 387 attributes of `omena`, *apple*, and 140 attributes of `vero#uudistus`, *tax reform*, in the decreasing order of shared weight. The boxed material corresponds to my variables. The column sums are $1 - $ `Sim`, `PHead` and `PTail`; In this caption, `NShared` is the number of rows in the table; `NHead` and `NTail` refer to the full representations of the two words.

Shared	**Head**	**Tail**	**Attribute**
0.0237	0.0193	0.0296	`olla-subj-`
0.0151	0.0133	0.0173	`kuoria-obj-`
0.0125	0.0121	0.0130	`syödä-obj-`
0.0089	0.0085	0.0094	`kilo-mod-`
0.0082	0.0036	0.0289	`-attr-keittää`
0.0077	0.0048	0.0137	`-cc-porkkana`
0.0074	0.0060	0.0094	`-attr-kuoria`
0.0072	0.0072	0.0072	`myydä-obj-`
0.0067	0.0157	0.0036	`-attr-kotimainen`
0.0059	0.0036	0.0108	`-cc-sipuli`
0.0056	0.0048	0.0065	`saada-obj-`
0.0053	0.0085	0.0036	`-attr-iso`
0.0051	0.0060	0.0043	`ei-subj-`
0.0048	0.0036	0.0065	`kg-mod-`
0.0046	0.0048	0.0043	`-attr-raastaa`
...	...	...	...
0.0020	0.0012	0.0036	`soseuttaa-obj-`
0.0018	0.0024	0.0014	`tuonti-attr-`
0.0018	0.0024	0.0014	`tonni-mod-`
0.0018	0.0024	0.0014	`tehdä-obj-`
0.0018	0.0024	0.0014	`lohkoa-obj-`
0.0018	0.0024	0.0014	`-attr-tulla`
0.0018	0.0012	0.0029	`viljely-attr-`
0.0018	0.0012	0.0029	`leikata-obj-`
0.0018	0.0012	0.0029	`kuutioida-obj-`
...	...	...	...
0.0009	0.0012	0.0007	`-attr-mätä`
0.0009	0.0012	0.0007	`-attr-kuutioida`
0.0009	0.0012	0.0007	`-attr-hauduttaa`
0.2712	0.2830	0.3642	

Table 4.13: The weights of the 105 shared attributes of 387 attributes of `omena`, *apple*, and 693 attributes of `peruna`, *potato*, in the decreasing order of shared weight. The boxed material corresponds to our variables. The column sums are $1 - $ `Sim`, `PHead` and `PTail`. In this caption, `NShared` is the number of rows in the full table; `NHead` and `NTail` refer to the the full representations of the two words.

Table 4.14: The rank variables for the sub-sample of 16 pairs, approximately in the form of a data frame. A high unknown `Knar` is actually `1000000` in these experiments.

Head	Tail	Rank	Knar
leivonnainen	kakku	3	34
päättäjä	luottamus#henkilö	8	2
oikeus#käytäntö	laki	8	>100
strategia	ohjelma	9	>100
isku#ryhmä	jury	8	>100
velka#kirja	omaisuus	12	>100
yhtiö#kokous	istunto	12	21
palvelu#työn#antaja	työn#antaja#liitto	7	1
komeus	ikä	15	>100
asunto#tuotanto	väki#luku	4	>100
perus#rakenne	perus#asia	8	>100
storgårds	ekki	20	>100
vasta#kaiku	super#bingo	16	6
sankar	asia	20	>100
ramppi	asunto	15	>100
hitunen	disk#ontto#korko	18	>100

Sub-sample examples Table 4.14 lists the values of `Rank` and `Knar` for these 16 pairs. The `Rank` values are all small, since the data consists of pairs with `Rank` at most 20. Some of the known `Knar` values exceed that limit, and most exceed even the limit of 100, beyond which all we know is that the true `Knar` is somewhere above 100, up to the vocabulary size.

Table 4.15 shows the distributional variables for this sub-sample of 16 pairs.

Table 4.15: The distributional variables for the sub-sample of 16 pairs: information radius, numbers of attributes, and proportions of shared weight.

Head	Tail	Sim	NShared	NHead	NTail	PHead	PTail
leivonnainen	kakku	0.7930	44	201	314	0.2384	0.2476
päättäjä	luottamus#henkilö	0.6446	65	732	253	0.4139	0.4571
oikeus#käytäntö	laki	0.7209	87	133	4817	0.7245	0.2071
strategia	ohjelma	0.6667	336	768	4835	0.6179	0.3918
isku#ryhmä	jury	0.7749	22	127	229	0.2548	0.2475
velka#kirja	omaisuus	0.7627	55	131	1324	0.5550	0.1840
yhtiö#kokous	istunto	0.6995	117	461	678	0.4287	0.3412
palvelu#työn#antaja	työn#antaja#liitto	0.7100	27	106	195	0.4745	0.2506
komeus	ikä	0.8126	20	100	1347	0.4071	0.1315
asunto#tuotanto	väki#luku	0.7595	29	208	249	0.2153	0.3790
perus#rakenne	perus#asia	0.7646	20	161	223	0.2491	0.2527
storgårds	ekki	0.7952	7	98	33	0.1454	0.3333
vasta#kaiku	super#bingo	0.6416	2	63	5	0.4279	0.3171
sankar	asia	0.8071	169	432	8018	0.5371	0.1952
ramppi	asunto	0.7630	62	224	2742	0.3644	0.2120
hitunen	disk#ontto#korko	0.8541	7	148	45	0.0843	0.3600

4.5 A look at the distributional variables

In this section, I examine the inventory of distributional variables in light of the sample of 400 pairs and my intuitive classification of them into `good` and `bad`. My objective is to determine whether or not they might be of any use in predicting the classification.

Figure 4.1 displays some different ways to visualise such pairs of distributions. There are two pairs of histograms: the upper two histograms show the actual counts of the head–tail pairs with `Sim` in a certain range, with total areas of 228 and 172; the lower two histograms show the proportions of those head–tail pairs of all 400, each with total area normalised to 1.0. The shapes of the count and density histograms are identical. Only the scale of the vertical axis is different.

Below the histograms, a single panel contains the smoothed versions of the two density histograms above it. I fit these curves to the underlying data vectors with the R function `density`. Since I used the function with its default parameters, the fits tend to leak a little outside the real range of the variables.

The right side of Figure 4.1 shows three bar graphs. Each of them contains the bars whose heights indicate the number or density of good and bad pairs with their `Sim` in a certain range.

I choose to use the pairs of the smoothed density curves. These seem to provide the clearest picture of the relative location and overlap of the underlying data sets.

When plotting the densities, the hope is that the overall positions of the two curves differ. Otherwise the variable can not predict the classification. It may also be expected that this difference is in a natural direction for some variables:

- For `Sim`, the good pairs should be located to the left of the bad pairs, because higher values of `Sim` indicate less similarity.
- For `NShared`, the good pairs should be located to the left of the bad pairs, because more shared attributes should indicate more higher similarity.

But the two curves have a large overlap.

4.5.1 The similarity score

First, let us examine the distribution of `Sim`, which is the information radius. Remember that the data set consists of head–tail pairs with a relatively low `Sim`.

Different ways to visualise the distributions of a variable separately for good pairs and bad pairs

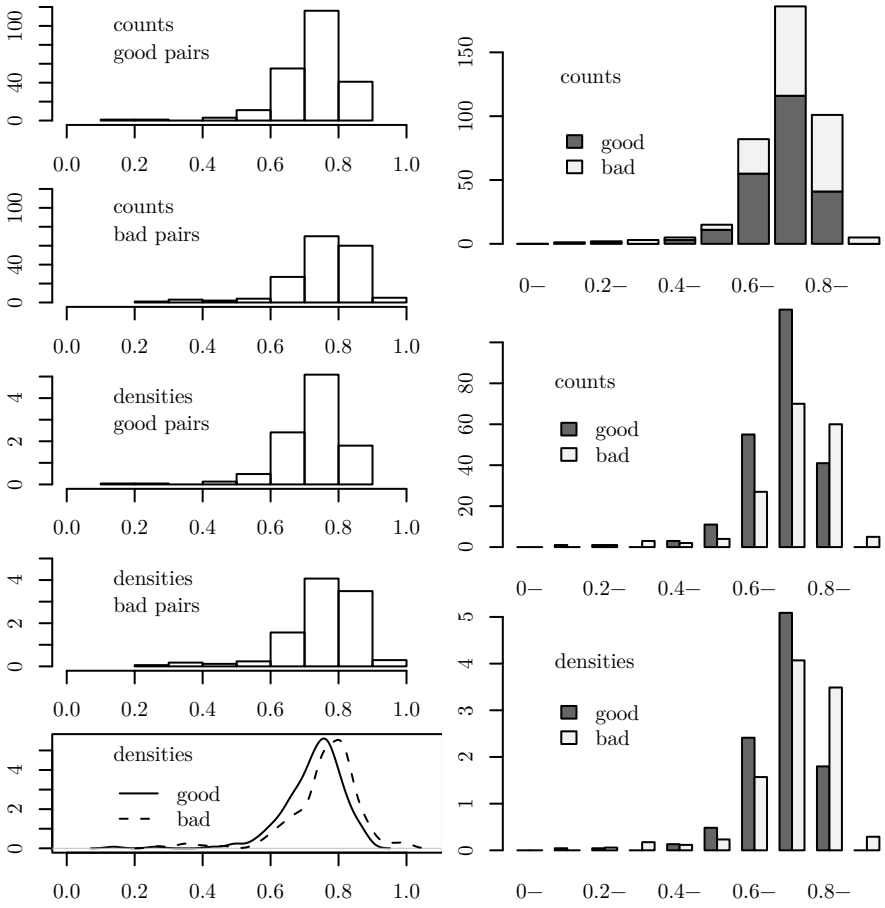


Figure 4.1: Ways to visualise the distributions of the the values of a variable (here, `Sim`) for good pairs and bad pairs. To the left, two frequency histograms, then two histograms of proportions, and finally two estimated density curves over each other. To the right are the two bar graphs of counts and one bar graph of proportions. The density curves were selected with the solid line marking the distribution for the good pairs and the dashed line for the bad pairs. (The variable here is Sim, so the horizontal axis covers its range $[0\,..\,1]$, with a little room for the density estimates to leak into.)

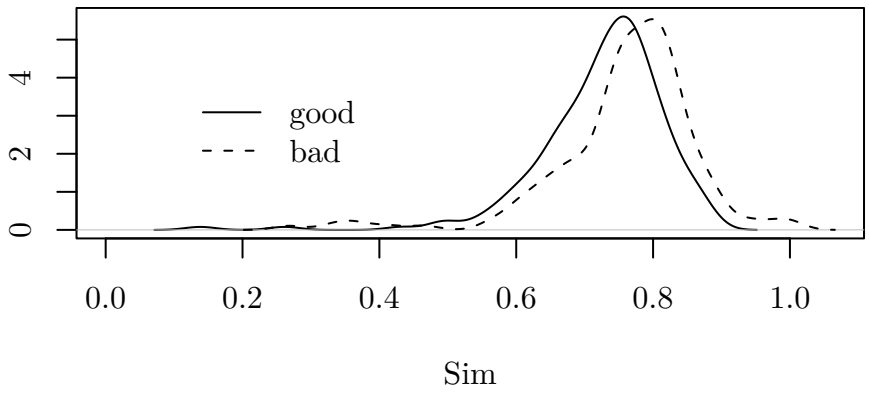


Figure 4.2: Smoothed densities of `Sim` for the pairs in the sample of 400 that I judged to be semantically similar (`Sense = good`, solid curve) and those that I judged to not be semantically similar (`Sense = bad`, dashed curve). A smaller `Sim` indicates greater distributional similarity, so the difference of location is in the expected direction, but the overlap is great.

Figure 4.2 shows the estimated density curves of `Sim` for the pairs with `Sense = good` and for the pairs with `Sense = bad`. This is actually the density pair plot from Figure 4.1. (The leakage of the dashed curve beyond 1.0 does not mean that such pairs occur in the data set. It is simply a result of the smoothing method.)

It is reassuring to see that the locations of the two curves differ in the expected direction: the information radius is, on average, higher for a bad pair than for a good pair.

Still, the distributions of `Sim` overlap much for the two semantic classes. This is not surprising, given the initial observation that distributional similarity is not the same as semantic similarity. In addition, these pairs are, by construction, distributionally relatively similar, yet I classified a large proportion of them as not being semantically similar.

Figure 4.3 shows similar density pairs (`Sim` for `good` and `bad`) separately for the part of the sample in which I thought I was `sure` of my classification, and for the part in which I thought I was `unsure`. These do not seem to differ much from each other or from the whole data set, so I will not focus on this distinction.

Finally, Figure 4.4 compares the densities of `Sim` for the pairs I was `sure` about and for the pairs I was `unsure` about. No difference of location occurs here. The greater width of the solid curve can be attributed to the larger data sets, 265 pairs, with 135 under the dashed curve. I did not expect `Sim`

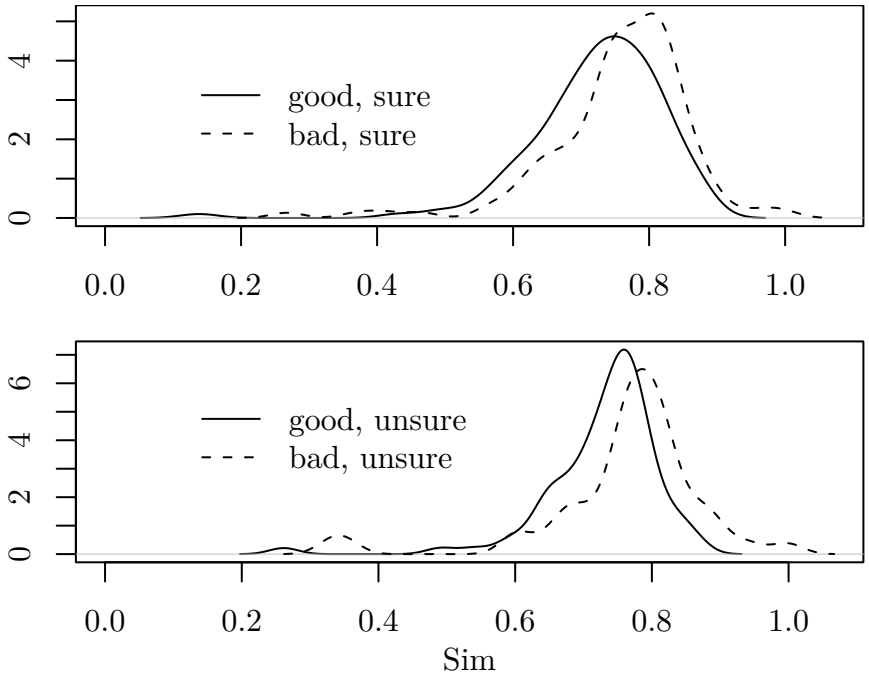


Figure 4.3: The upper diagram depicts the densities of `Sim` by `Sense` for those pairs that I found easy to classify (`Ease` = `sure`) as good or bad (semantically similar or not). The lower diagram depicts the same for those pairs that I found difficult to classify (`Ease` = `unsure`). Both diagrams show both the expected direction of location and the extensive overlap.

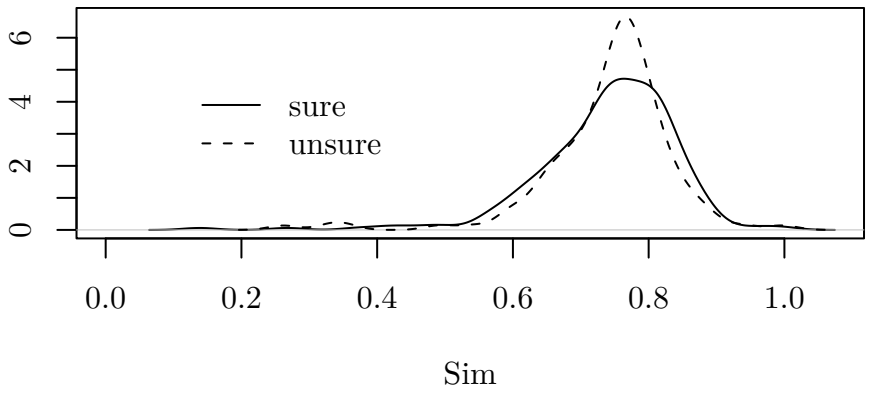


Figure 4.4: Smoothed densities for `Sim` for the pairs in the sample of 400 that I found easy to classify (`Ease` = `sure`, solid curve), and for those that I found difficult (`Ease` = `unsure`, dashed curve). These show no difference in location. This is not a problem since `Sim` was never expected to be a measure of the ease of classification.

to measure the ease of classification, and it does not appear to mesure it.

To sum up, `Sim` separates semantically the similar pairs from the dissimilar pairs, in a set of head–tail pairs where the tail is distributionally relatively similar to the head, the right way around but not very strongly.

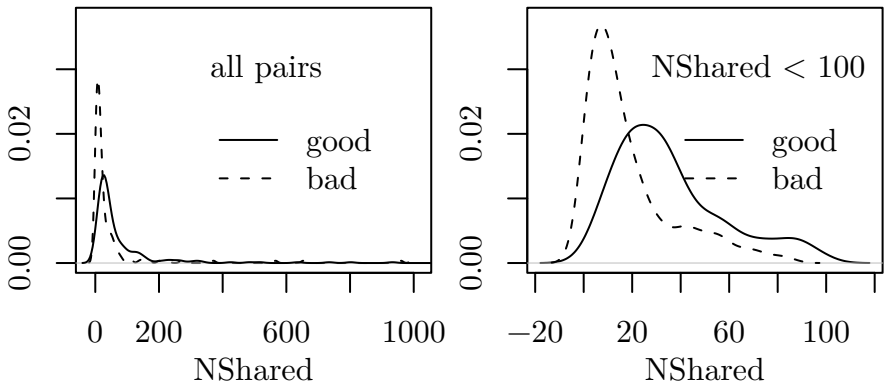


Figure 4.5: Smooth densities of `NShared` for the pairs I judged to be semantically similar (`good`, solid) and not similar (`bad`, dashed curves). The left diagram, for the full sample of 400 pairs, shows mainly a long tail. The right diagram is for the pairs with small or moderate `NShared` only and magnifies the more interesting shape at the extreme left.

4.5.2 Numbers of attributes

Let us now turn to the variables that count attributes: `NShared`, `NHead` and `NTail`. Of these, `NShared` is the most interesting: it alone might be expected to be a simple measure of distributional similarity. The other two limit it in a natural way, and their ratio to it might be of interest.

Figure 4.5 shows the densities of `NShared` for my good and bad pairs. There is a long tail with a few pairs with a large `NShared`, mostly `good`, that makes it difficult to see the shape for most of the data, so a second diagram shows the densities for the subset with a small `NShared`. It looks as if a very small `NShared` might be a warning sign, as I expected it to be.

Tables 4.16 and 4.17 list the pairs having only a couple of shared attributes. The `Sense` and `Ease` columns show clearly that I have classified almost all of them as `bad` and that I have been `sure` of that classification.

Table 4.16: The pairs in the sample of 400 that share less than two of their attributes, as seen in the column for `NShared`. Most are `bad`, and I was `sure` of most of the classifications.

Head	Tail	Sense	Ease	Rank	Knar	NShared	NHead	NTail	PHead	PTail
emu-#jäsenyys	sääst	bad	sure	6	31	1	289	1	0.1563	1.0000
esitys#lista	suunnite	good	unsure	11	>100	1	210	23	0.1216	0.9355
etelä#kaakko	valta#meri	bad	sure	12	>100	1	3	151	0.2500	0.0078
fenno	rahasto	bad	unsure	11	>100	1	5	1263	0.4286	0.0003
kattaus	päivä#lämpö#tila	bad	sure	17	>100	1	73	35	0.1031	0.6473
lokki	metsä#palo#varoitus	bad	sure	9	>100	1	151	8	0.0656	0.9357
maku#asia	metsä#palo#varoitus	bad	sure	15	33	1	22	8	0.3415	0.9357
myyrmäkitalo	suunnite	bad	sure	20	>100	1	55	23	0.0508	0.9355
palata	kukka	bad	sure	20	>100	0	2	783	0.0000	0.0000
ralli#autoilu	a-poiki	bad	unsure	15	13	1	83	10	0.0732	0.5396
skaala	maksimi#lämpö#tila	bad	sure	7	>100	1	149	3	0.1386	0.9971
sääli	op-pirkka	bad	sure	20	100	1	75	1	0.2083	1.0000
tilaus#kanta	uskominen	bad	sure	13	61	1	181	20	0.2674	0.8039
timjami	leivin#jauhe	good	sure	3	2	1	58	23	0.2362	0.6435
viides#osa	maksimi#lämpö#tila	bad	sure	11	>100	1	198	3	0.1502	0.9971

Table 4.17: The pairs in the sample of 400 that share only two or three of their attributes. All are bad, and I was sure of all but one of the classifications.

Head	Tail	Sense	Ease	Rank	Knar	NShared	NHead	NTail	PHead	PTail
for	balansor	bad	sure	8	67	2	512	3	0.0404	0.9934
hertta#ässä	varas#lähtö	bad	sure	12	4	3	60	70	0.3274	0.2419
maailman#lista	tukko	bad	sure	9	73	3	106	44	0.2085	0.6070
rang	jets	bad	unsure	3	6	2	11	144	0.5776	0.0884
realisti	loppu#ilta#päivä	bad	sure	11	13	3	114	42	0.2613	0.4242
sankaritar	vasta#väittäjä	bad	sure	5	>100	3	121	35	0.1149	0.9830
spiri	päivä#työ	bad	sure	8	>100	2	31	114	0.0444	0.1828
talvi#olympialainen	luonto#kuva	bad	sure	17	24	3	110	91	0.1729	0.3441
tasaisuus	suosittuja	bad	sure	14	>100	3	126	16	0.1528	0.8140
tavata	sanan#valta	bad	sure	9	>100	3	16	103	0.1579	0.1337
urheilu-#uutinen	suku#kokous	bad	sure	3	8	2	11	88	0.2500	0.3374
urheilu-#uutinen	-#tilanne	bad	sure	5	5	2	11	62	0.2500	0.2323
vaaka#lauta	näkö#piiri	bad	sure	4	5	2	17	50	0.7143	0.7531
vasta#kaiku	super#bingo	bad	sure	16	6	2	63	5	0.4279	0.3171

Figure 4.6 shows the smoothed densities of `NHead` and `NTail` for my good and bad pairs, with a magnifying diagram for the pairs in which the number of attributes is at the most moderate.

The pairs having one word with a very small number of attributes seem to be bad, but this is reflected as a small `NShared`.

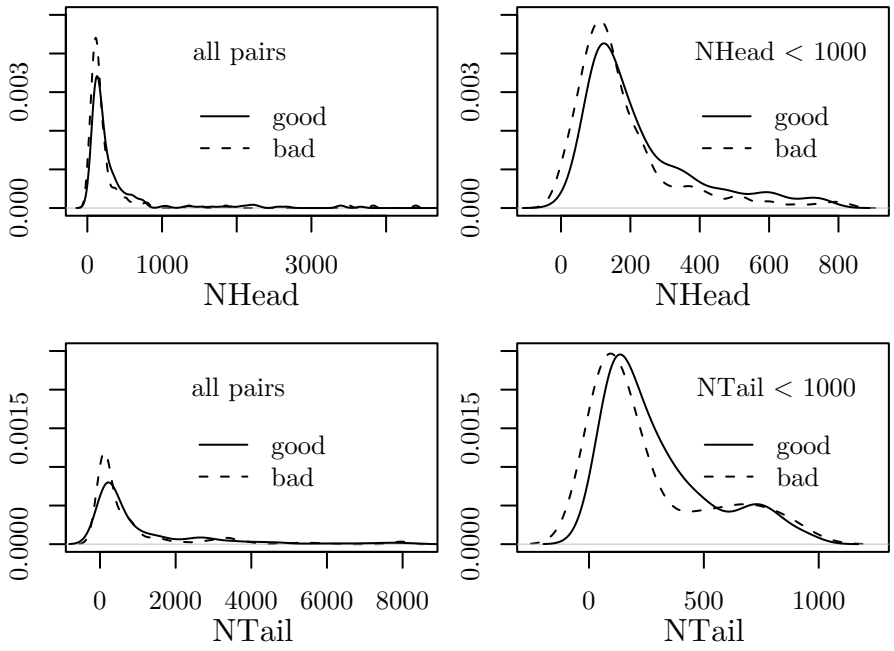


Figure 4.6: Smooth histograms (density estimates) for `NHead` and `NTail` by `Sense` levels (solid line for `good`, dashed for `bad`). The diagrams for all show only the long tail to the right, because a few words have a very high number of attributes. The diagrams for a moderate number of attributes magnify the interesting shapes at the extreme left.

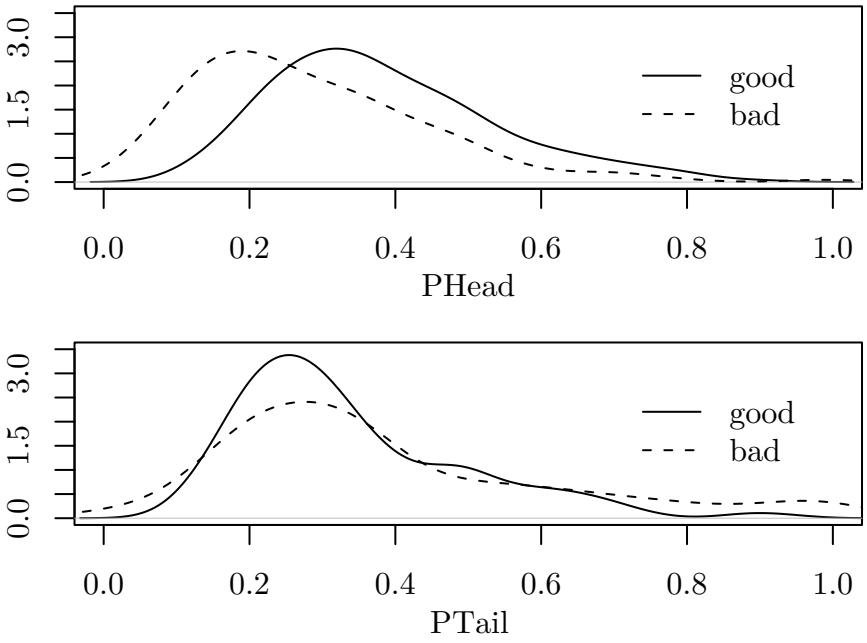


Figure 4.7: Smoothed densities of `PHead` (upper panel) and `PTail` (lower panel) for those pairs in the sample of 400 that I judged to be semantically similar (`Sense` = `good`, solid curves) and those that I judged to not be semantically similar (`Sense` = `bad`, dashed curves). While `PHead` seems usable for predicting the classification, `PTail` does not.

4.5.3 Proportions of shared weight

The shared proportions of probability, `PHead` and `PTail`, seem to offer a mild separation of the good from the bad in `PHead` and a puzzling asymmetry between `PHead` and `PTail`. The overlapping density estimates are displayed in Figure 4.7, where `PHead` looks somewhat promising, while `PTail` has `good` and `bad` at the same location.

4.6 Recursive partitioning

With my intuitive judgements at hand, I used the recursive partitioning library `rpart` in R to train classification trees and to see how, and how well, they can match my semantic intuition. I will describe the steps roughly in the order that I actually progressed. First I trained a few trees on the full sample of 400 pairs, then I redid them on the `sure` pairs only, and finally I tested them on a separate test set of 100 pairs.

The method 'An introduction to R' (available from R web site) states the following about **tree-based models**:

> ... tree-based models seek to bifurcate the data, recursively, at critical points of the determining variables in order to partition the data ultimately to into groups that are as homogeneous as possible within, and as heterogeneous as possible between. The results often lead to insights that other data analysis methods tend not to yield.

Therneau and Atkinson (1997) describe the tree-building procedure as follows with their emphasis and my ellipses:

> ... first the single variable is found which best splits the data into two groups ... The data is separated and then this process is applied *separately* to each sub-group and so on recursively until the subgroups either reach a minimum size ... or until no improvement can be made.
>
> The resultant model is, with certainty, too complex, and the question arises as it does with all stepwise procedures of when to stop. The second stage of the procedure consists of using cross-validation to trim back the full tree. ...

Three model formulas, two training sets The following subsections set up three different model formulas. The corresponding models are trained on all 400 pairs. For discussion, these models are referred to as RANK/ALL, COUNT/ALL, and RATIO/ALL.

The first part of each model name refers to the model formula. The second part refers to the training set: I trained another three models, RANK/SURE, COUNT/SURE, and RATIO/SURE, by applying the same model formulas but using only the 265 `sure` pairs as data.

Two of the models have the best success rates for five different prediction tasks on the test pairs.

4.6.1 A model with all variables

Figure 4.8 displays a classification tree whose model formula says to predict `Sense` using any of my repertoire of the input variables from Section 4.4:

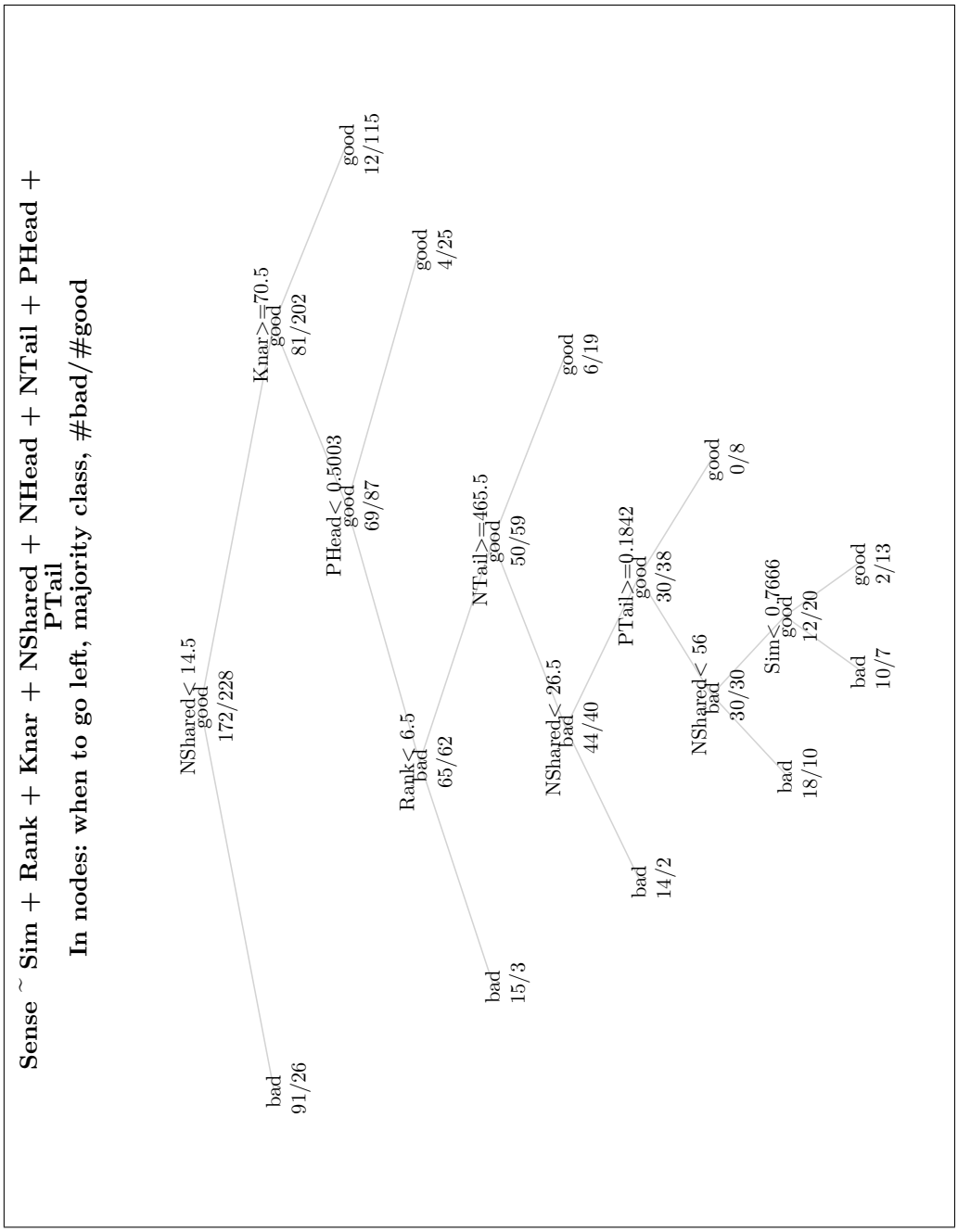
```
Sense ~ Sim + Rank + Knar + NShared + NHead + NTail +
                    PHead + PTail
```

Training this model with all 400 pairs produced the first model, which I call RANK/ALL. The formula part of the name refers to the presence of the 'ranking' variables `Sim`, `Rank`, and `Knar` in the formula. Figure 4.8 displays this model graphically.

To understand the meaning and use of the classification tree, let us trace the steps of classification for the anecdotal `Head`–`Tail` pairs. (I began with `omena`–`appelsiini`, which should be good. Then I found the surprising tail words `vero#uudistus` and `lanka` high on the list of `omena`, and I would like to identify them as bad. I also added the good tail word `peruna` from the list of `omena`.) See Table 4.11 on page 123 for their data frame. For each pair, I now also note whether the final classification by the model was *correct* or *incorrect*, that is, whether it matched my semantic intuition.

- This model classified `omena`–`appelsiini` *correctly* as `good` through the following sequence of binary decisions:
  1. This pair has `NShared == 30`, so the condition `NShared < 14.5` in the root node is false: take the right branch.
  2. This pair has `Knar == 6`, so the condition `Knar >= 70.5` in the node is false: take the right branch.
  3. We are in a leaf node: classify the pair according to the majority class in this node – `good`.
- This model classified `omena`–`vero#uudistus` *correctly* as `bad` through the following sequence of binary decisions:
  1. This pair has `NShared == 14`, so the condition `NShared < 14.5` in the root node is true: take the left branch.
  2. We are in a leaf node: classify the pair according to the majority class in the node – `bad`.
- This model classified `omena`–`lanka` *incorrectly* as `good` based on the sequence `NShared == 24 >= 14.5`, then `Knar > 100 >= 70.5`, then `PHead == 0.38 < 0.5003`, then `Rank == 12 >= 6.5`, then `NTail == 374 < 465.5`, ending at a leaf node containing a minority of 6 `bad` and a majority of 19 `good` pairs.

Figure 4.8: The RANK/ALL model, which predicts `Sense` from all input variables, including `Sim`, `Rank` and `Knar`. This model is trained on all 400 pairs and has an overall success rate of 82% on them (Table 4.18 on page 149) and 87% on the 256 pairs I was `sure` about (Table 4.19 on page 150). Compare this to the same model trained on the `sure` pairs, found in Figure 4.12 on page 152. A textual form of this model is in Appendix C on page 194.



- This model classified `omena–peruna` *correctly* as `good` based on the sequence `NShared == 105 >= 14.5`, then `Knar == 16 < 70.5`, ending at a leaf node containing a minority of 12 `bad` and a majority of 115 `good` pairs.

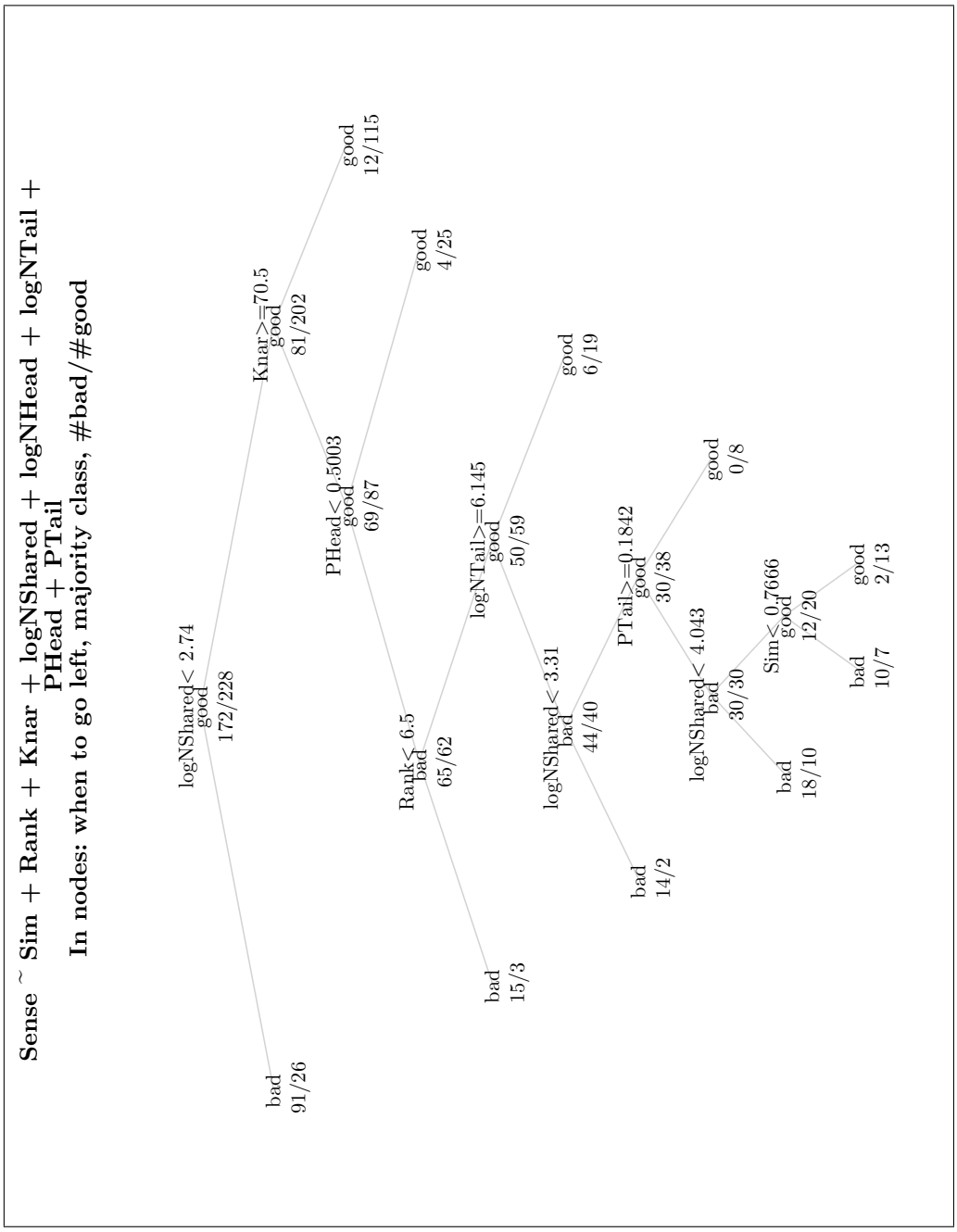
These models are not sensitive to skewed data Figure 4.9 displays a variant model with logarithmic counts, to counter a suspicion that the skewed variables would need to be so transformed for the models to work well. The model matches that of fIgure 4.8 exactly, and I shall pay no further attention to logarithmic transformations.

The three logarithmically transformed variables are best glossed with the simple formulas that define them:

```
logNShared    = log(1 + NShared)
logNHead      = log(1 + NHead)
logNTail      = log(1 + NTail)
```

The addition of 1 before taking the logarithm actually helps here, since the sample contained a pathological case or two in which a count was 0.

Figure 4.9: This is the same as Figure 4.8 but with the three highly skewed count variables transformed by $x \mapsto \log(1+x)$. This suggests that recursive partitioning is not sensitive to the skewness of the data.



4.6.2 A model without ranking variables

The classification tree in Figure 4.10 is based on the word representations only, without access to the ranking table variables `Sim`, `Rank` and `Knar`. The model formula uses `NShared`, `NHead`, `NTail`, `PHead` and `PTail` to predict `Sense`:

```
Sense ~ NShared + NHead + NTail + PHead + PTail
```

The formula part of the name COUNT/ALL refers to the variables `NHead` and `NTail` that *count* the attributes of `Head` and `Tail`. The third model formula replaces these two with variables that relate these counts to `NShared`.

Let us note the four anecdotal results with no more tracing of the steps. See Table 4.11 on page 123 for the variables.

- This model classified `omena`–`appelsiini` *correctly* as `good`.
- This model classified `omena`–`peruna` *correctly* as `good`.
- This model classified `omena`–`vero#uudistus` *correctly* as `bad`.
- This model classified `omena`–`lanka` *incorrectly* as `good`.

4.6.3 Another model without ranking variables

The classification tree for RATIO/ALL in Figure 4.11 uses the two derived ratios `HShared` and `TShared` instead of the underlying counts `NHead` and `NTail`. These derived variables are best glossed by the simple formulas that define them:

```
HShared    = NShared / NHead
TShared    = NShared / NTail
```

The model formula for RATIO/ALL is:

```
Sense ~ NShared + HShared + TShared + PHead + PTail
```

Let us note the four anecdotal results. See Table 4.11 on page 123 for the variables.

- This model classified `omena`–`appelsiini` *correctly* as `good`.
- This model classified `omena`–`peruna` *correctly* as `good`.
- This model classified `omena`–`vero#uudistus` *correctly* as `bad`.
- This model classified `omena`–`lanka` *incorrectly* as `good`.

Figure 4.10: The COUNT/ALL model, which predicts `Sense` without the information radius or ranks, from the attribute counts and proportions only. This model is trained on all 400 pairs and has an overall success rate of 82% for them (Table 4.18 on page 149) and 86% on the 256 pairs that I was `sure` about (Table 4.19 on page 150). Compare these results to the same model trained on the `sure` pairs, Figure 4.13 on page 153. (A textual form of this model is in Appendix C on page 195.)

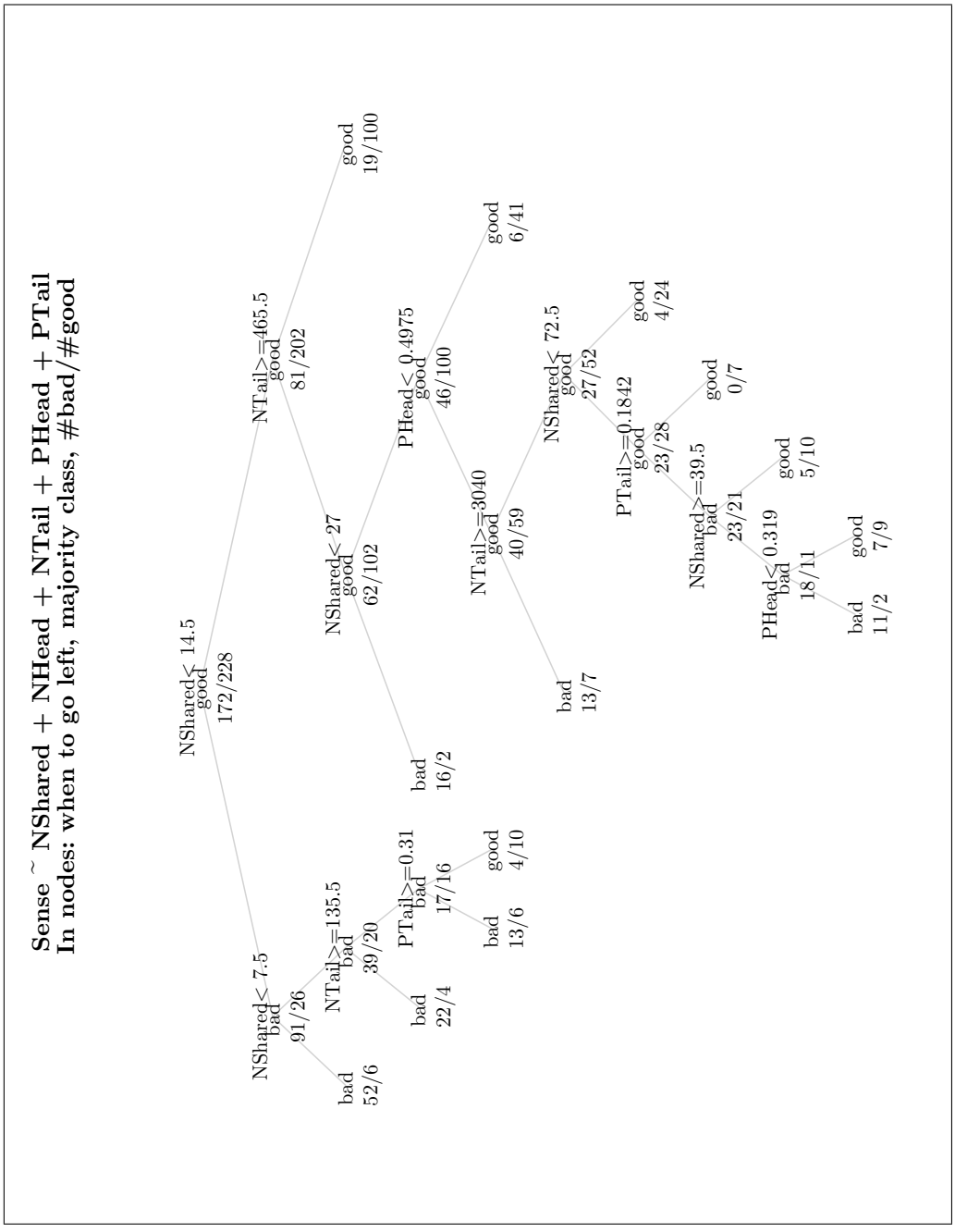
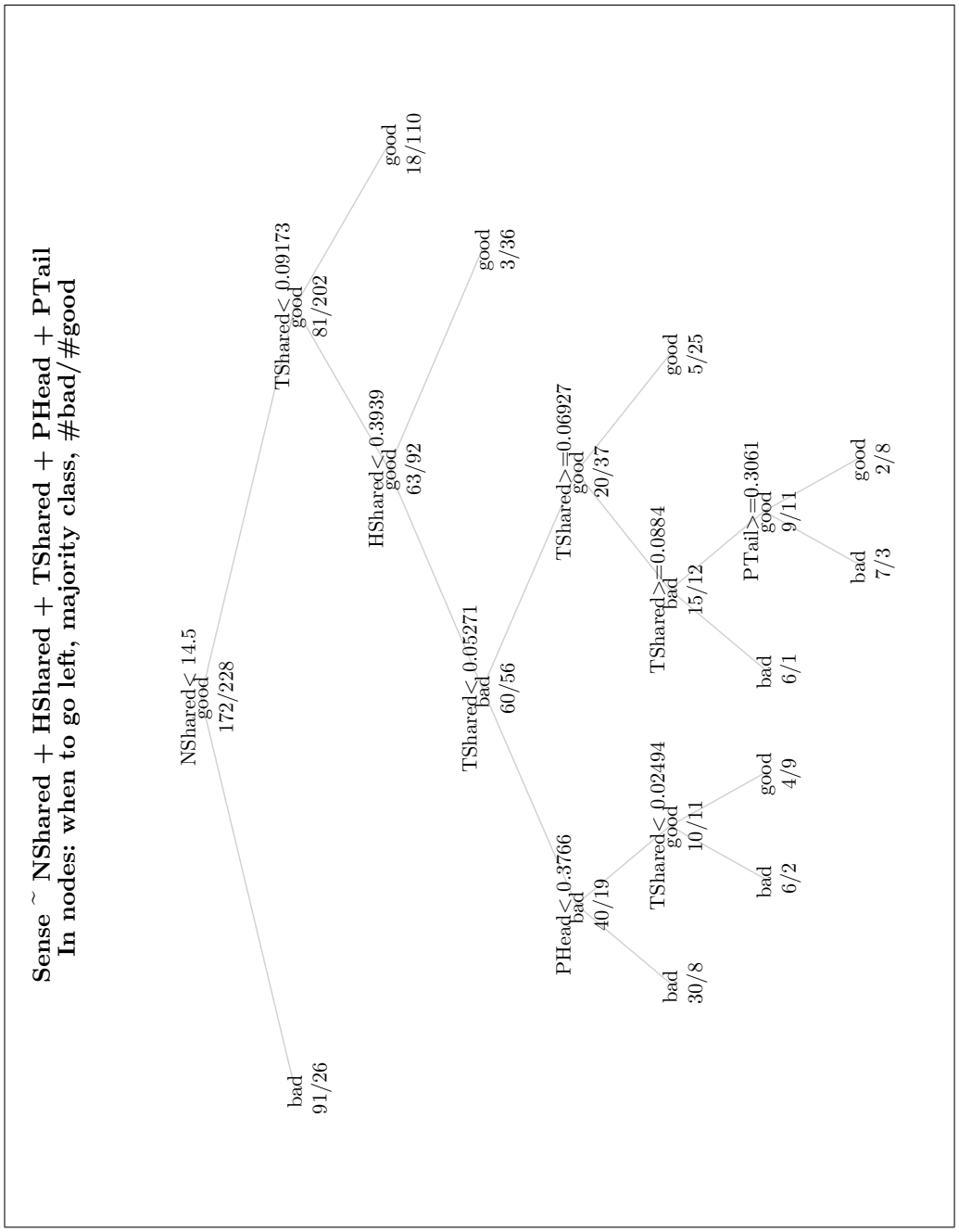


Figure 4.11: The RATIO/ALL model, which predict `Sense` using the ratios of `NShared` to `NHead` and `NTail`. This model is trained on all 400 pairs and has an overall success rate of 82% for them (Table 4.18 on page 149) and 86% on the 256 pairs that I was `sure` about (Table 4.19 on page 150). Compare these to the same model trained on the `sure` pairs, found in Figure 4.14 on page 154. A textual form of this model is in Appendix C on page 196.



Preliminary observation The anecdotal tests with the four tail words of `omena`, *apple*, gave the same result on all three models so far: `appelsiini`, *orange*, and `peruna`, *potato*, were classified correctly as `good`, and `vero#uudistus`, *tax reform*, correctly as `bad`, but `lanka`, *thread*, incorrectly as `good`. The background is that both `vero#uudistus` and `lanka` share one dominating attribute with `omena`, namely the modifier `vihreä`, *green*, which was inflated for `omena` by the presence of a theatre group called `Vihreä Omena`, *Green Apple*. Behind `lanka`, there is a magazine called `Vihreä lanka`. The two good words, on the other hand, share several appropriate attributes with `omena`.

4.6.4 Success rates on training data

We can get a better idea of the goodness of each classification tree by observing how often it classifies the pairs correctly. The success rates here refer to the training pairs and should therefore be too good. A separate hand-classified test set will be presented later.

However, the training method does use cross-validation to prune the trees (Therneau and Atkinson, 1997). Here is the relevant part of the quote again:

> The resultant model is, with certainty, too complex, and the question arises as it does with all stepwise procedures of when to stop. The second stage of the procedure consists of using cross-validation to trim back the full tree. ...

At least five different success rates are of interest. I classified the pairs intuitively to be `good` and `bad`, so those are the *expected* results. The classification trees also classify the pairs to be `good` and `bad`; these are the *predicted* results, based on the input variables of the classifier.

The predicted result is *correct* when it is the same as the expected result. This definition leads to the four correctness rates in the margins of the contingency tables such as those in Table 4.18, and a fifth success rate for each table as a whole.

Table 4.19 shows the success rates for the same models, trained on all 400 pairs, when they are used to predict the `Sense` of those pairs that I considered easy to classify.

Table 4.18: Success rates of RANK/ALL, COUNT/ALL and RATIO/ALL on all 400 pairs. This is exactly the training data of these models. Compare these to the success rates on the 256 `sure` pairs (Table 4.19) and to the same model formulas when trained on the `sure` pairs (Table 4.20 on page 155 and Table 4.21 on page 156).

Successes of RANK/ALL (Figure 4.8) on all training pairs

	Predicted `bad`	Predicted `good`	Rate
Expected `bad`	**148**	24	0.86
Expected `good`	48	**180**	0.79
Rate	0.76	0.88	

Overall success rate 0.82

Successes of COUNT/ALL (Figure 4.10) on all training pairs

	Predicted `bad`	Predicted `good`	Rate
Expected `bad`	**127**	45	0.74
Expected `good`	27	**201**	0.88
Rate	0.82	0.82	

Overall success rate 0.82

Successes of RATIO/ALL (Figure 4.11) on all training pairs

	Predicted `bad`	Predicted `good`	Rate
Expected `bad`	**140**	32	0.81
Expected `good`	40	**188**	0.82
Rate	0.78	0.85	

Overall success rate 0.82

Table 4.19: Success rates of RANK/ALL, COUNT/ALL and RATIO/ALL on the 265 `sure` pairs. These were included in the training data. Compare these to the success rates on all 400 pairs, Table 4.18.

Successes of RANK/ALL (Figure 4.8) on `sure` training pairs

	Predicted `bad`	Predicted `good`	Rate
Expected `bad`	**110**	15	0.88
Expected `good`	19	**121**	0.86
Rate	0.85	0.86	

Overall success rate 0.87

Successes of COUNT/ALL (Figure 4.10) on `sure` training pairs

	Predicted `bad`	Predicted `good`	Rate
Expected `bad`	**98**	27	0.78
Expected `good`	10	**130**	0.93
Rate	0.91	0.83	

Overall success rate 0.86

Successes of RATIO/ALL (Figure 4.11) on `sure` training pairs

	Predicted `bad`	Predicted `good`	Rate
Expected `bad`	**106**	19	0.85
Expected `good`	19	**121**	0.86
Rate	0.85	0.86	

Overall success rate 0.86

4.6.5 Training the models on the `sure` pairs only

My secondary classification of the 400 sample pairs can also be used to train a model with only the pairs I was sure about. This produces the models RANK/SURE (Figure 4.12), COUNT/SURE (Figure 4.13), and RATIO/SURE (Figure 4.14).

These models appear to be more simple than their counterparts that used all of the training pairs. It is tempting to think this could be due to the better quality of the training data, but the amount of training data is also smaller, and that might be the explanation.

These trees classify our four anecdotal tails of `omena`, in the order of `appelsiini`, `lanka`, `peruna`, and `vero#uudistus`, as follows.

The RANK/SURE model is *correct* on all four pairs. The classifications are as follows:

- It classifies `omena`–`appelsiini` *correctly* as `good`.
- It classifies `omena`–`lanka` *correctly* as `bad`.
- It classifies `omena`–`peruna` *correctly* as `good`.
- It classifies `omena`–`vero#uudistus` *correctly* as `bad`.

The COUNT/SURE model does the same as the models that were trained on all pairs.

- It classifies `omena`–`appelsiini` *correctly* as `good`.
- It classifies `omena`–`lanka` *incorrectly* as `good`
- It classifies `omena`–`peruna` *correctly* as `good`
- It classifies `omena`–`vero#uudistus` *correctly* as `bad`

The RATIO/SURE model goes wrong for `vero#uudistus`. This is the only one of the models that makes further decisions after determining that `NShared < 14.5`. This model is also incorrect for `appelsiini`.

- It classifies `omena`–`appelsiini` *incorrectly* as `bad`.
- It classifies `omena`–`lanka` *correctly* as `bad`
- It classifies `omena`–`peruna` *correctly* as `good`
- It classifies `omena`–`vero#uudistus` *incorrectly* as `good`

Figure 4.12: The RANK/SURE model predicts `Sense` using all variables. This model is trained on the 256 `sure` pairs and has an overall success rate of 77% on all 400 pairs (Table 4.20 on page 155) and 85% on the `sure` pairs (Table 4.21 on page 156). A textual form of this model is in Appendix C on page 197.

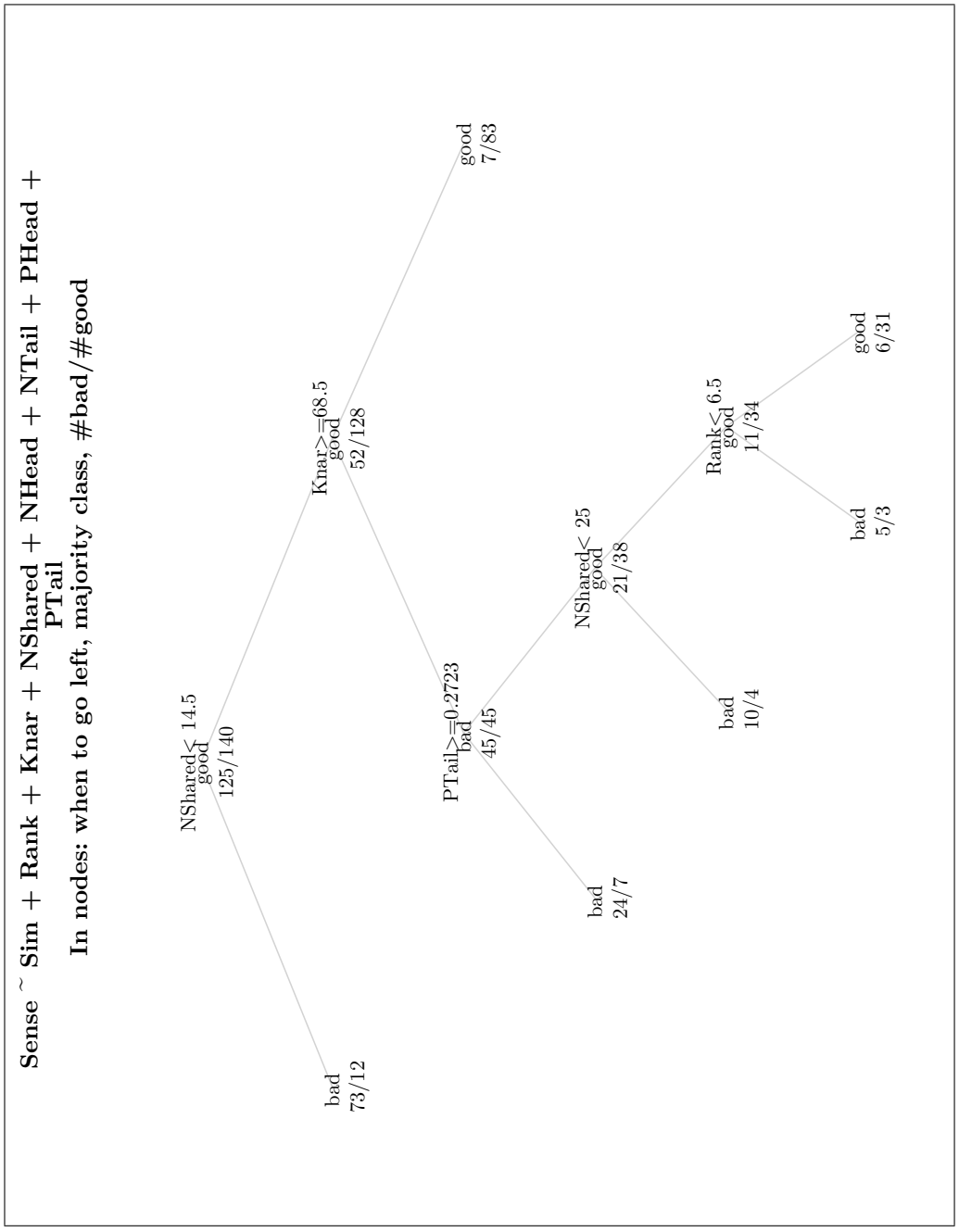


Figure 4.13: The COUNT/SURE model predicts `Sense` without ranking variables but including the count variables `NHead` and `NTail`. This model is trained on the 256 `sure` pairs and has an overall success rate of 78% on all 400 pairs (Table 4.20 on page 155) and 84% on the `sure` pairs (Table 4.21 on page 156). A textual form of this model is in Appendix C on page 198.

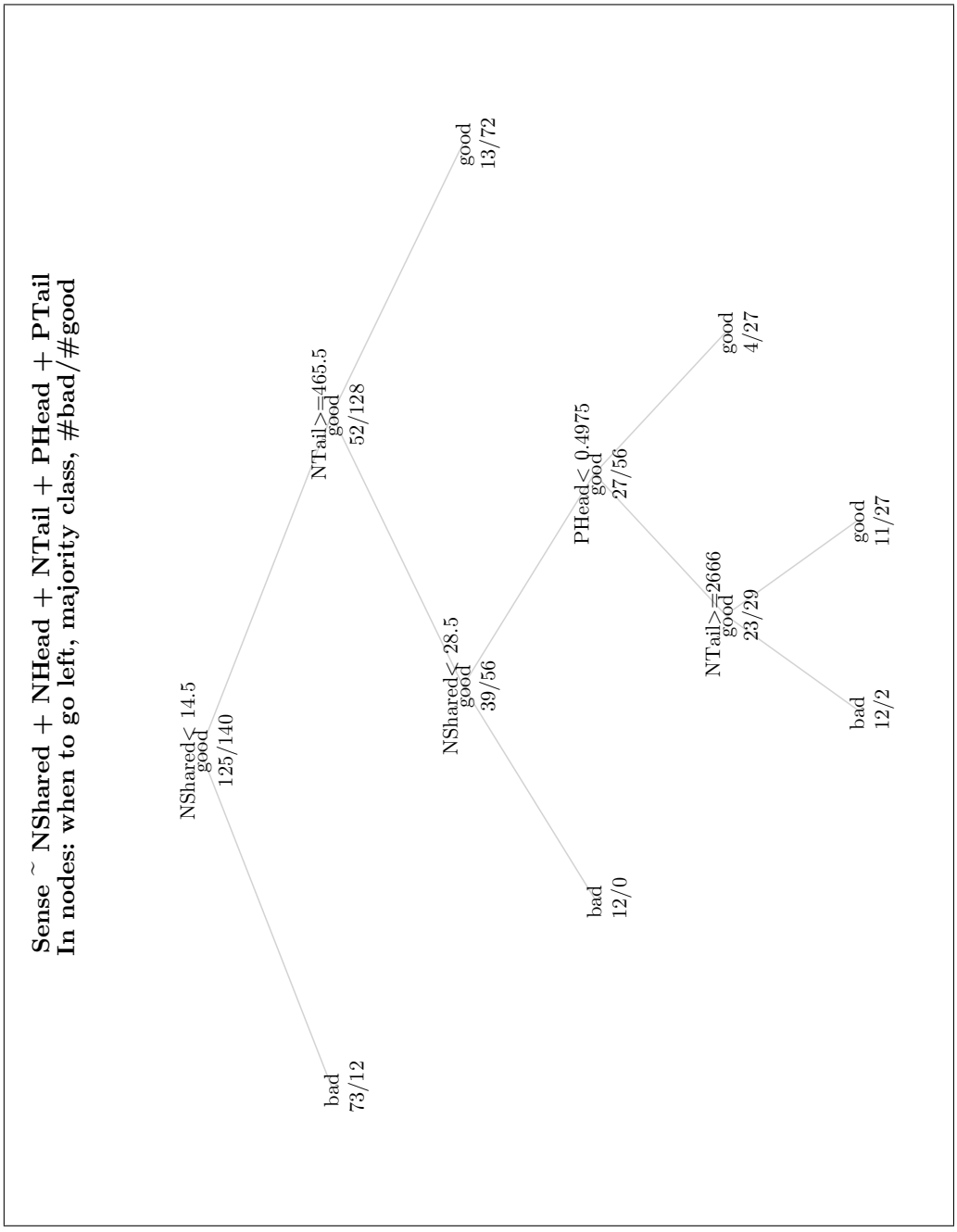


Figure 4.14: The RATIO/SURE model predicts `Sense` using the derived ratios `HShared` and `TShared` in place of `NHead` and `NTail` but trained on the `sure` pairs.

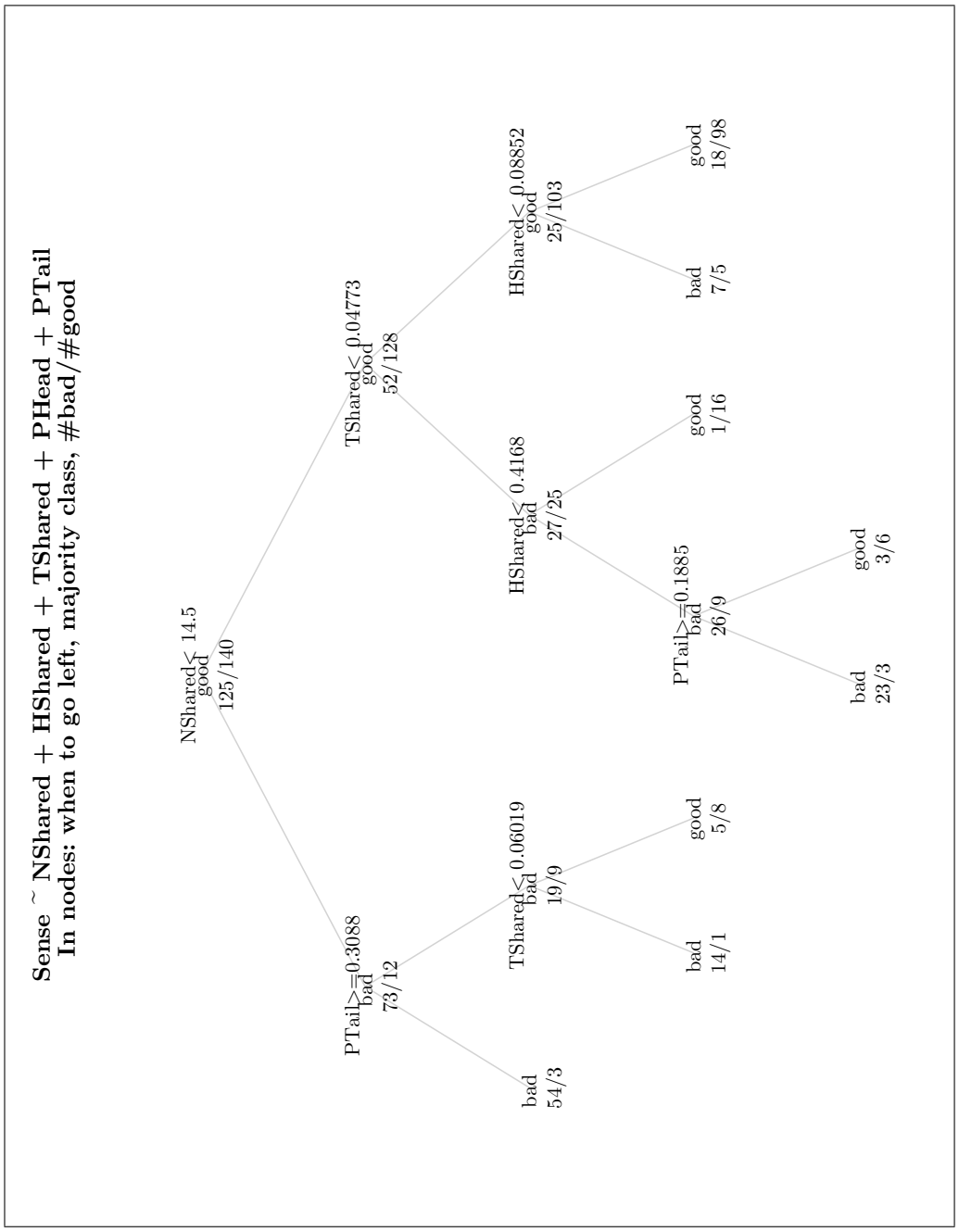


Table 4.20: Classification success rates of the three models trained on the 265 pairs that I was `sure` about, for all 400 training pairs. Success means the `Sense` that I *expected* (`good` or `bad`) is the same as the model *predicted*.

Successes of RANK/SURE (figure 4.12)

	Predicted `bad`	Predicted `good`	Rate
Expected `bad`	**143**	29	0.83
Expected `good`	62	**166**	0.73
Rate	0.70	0.85	

Overall success rate 0.77

Successes of COUNT/SURE (figure 4.13)

	Predicted `bad`	Predicted `good`	Rate
Expected `bad`	**122**	50	0.71
Expected `good`	39	**189**	0.83
Rate	0.76	0.79	

Overall success rate 0.78

Successes of RATIO/SURE (figure 4.14)

	Predicted `bad`	Predicted `good`	Correct
Expected `bad`	**121**	51	0.70
Expected `good`	36	**192**	0.84
Rate	0.77	0.79	

Overall success rate 0.78

4.6.6 Successes of the `sure` models on training pairs

Tables 4.20 and 4.21 display the five different success rates for the three classification trees trained on the `sure` pairs. In the first table, the predictions are computed for all 400 pairs. Here the test material actually contains pairs that were not in the training data for these models, but these pairs happen to be suspect for other reasons.

In the second table, the predictions are computed for the `sure` pairs, which are the training data for these models.

Table 4.21: Classification success rates of the three models trained on the 265 pairs I was `sure` about, for these 265 `sure` pairs. Success means the `Sense` that I *expected* (`good` or `bad`) is the same as the model *predicted*.

Successes of RANK/SURE (Figure 4.12)

	Predicted `bad`	Predicted `good`	Rate
Expected `bad`	**112**	13	0.90
Expected `good`	26	**114**	0.81
Rate	0.81	0.90	

Overall success rate 0.85

Successes of COUNT/SURE (Figure 4.13)

	Predicted `bad`	Predicted `good`	Rate
Expected `bad`	**97**	28	0.78
Expected `good`	14	**126**	0.90
Rate	0.87	0.82	

Overall success rate 0.84

Successes of RATIO/SURE (Figure 4.14)

	Predicted `bad`	Predicted `good`	Rate
Expected `bad`	**98**	27	0.78
Expected `good`	12	**128**	0.91
Rate	0.89	0.83	

Overall success rate 0.85

4.7 Success rates on test data

Since the performance of the models on their training data was not entirely disappointing, I made a similar but separate test set of one hundred more head–tail pairs to determine how well such models might be able to classify such pairs in general. The tail is again in the first twenty tails of the head, neither the head nor the tail contains any upper-case letters, and otherwise any head–tail pair in the full similarity table had the same probability of being included. (I used a different random seed than for the training set and tuned a threshold value by trial and error so that exactly one hundred pairs got through.)

This time, I knew from the start to classify the pairs fast. I spent only a few seconds on each pair, assigned it the `Sense` label of `good` or `bad` and the `Ease` label of `sure` or `unsure`, and never reconsidered any of my decisions. Defective pairs (not common nouns after all) were labeled as being `bad`. There are 67 `sure` pairs in this set, 31 of them `good` and 36 `bad`. All 100 test pairs are listed in Appendix B, after the 400 training pairs.

Input variables for the test data frame were simply extracted from the similarity matrix (`Sim`, `Rank`, `Knar`) and the word representations (`NShared`, `NHead`, `NTail`, `PHead`, `PTail`), or derived from these as before (`HShared`, `TShared`).

At this point I applied each of the six models to the test set and counted the successes. Table 4.22 summarises all five success rates of each model on the training data, and Table 4.23 on the test data. The latter are generally lower, as was to be expected, but still not wholly disappointing.

The best models One of the models stands out as the best in four of the success categories in Table 4.23: RATIO/SURE for being correct on the test pairs that I expected to be `good`, and on the test pairs that it predicted to be `bad`; RANK/SURE for being correct on the test pairs that I expected to be `bad`, and on the test pairs that it predicted to be `good`. In addition, RATIO/SURE wins the contest for overall success, though narrowly.

The model formula for RANK/SURE is

```
Sense ~ Sim + Rank + Knar + NShared + NHead + NTail +
                    PHead + PTail
```

and the model formula for RATIO/SURE is

```
Sense ~ NShared + HShared + TShared + PHead + PTail
```

where `HShared` is `NHead/NShared` and `TShared` is `NTail/NShared`. Both models were trained on the pairs of whose classification I was `sure`. Finally,

Table 4.22: Classification success rates of the six models for all training pairs, and for the training pairs that I was `sure` about. Success means the `Sense` I *expected* (`good` or `bad`) is the same as the model *predicted*. Table 4.23 displays the success rates for the test pairs.

Model	On all	On `sure`	Model	On all	On `sure`
Overall success rates					
RANK/ALL	0.82	0.87	RANK/SURE	0.77	0.85
COUNT/ALL	0.82	0.86	COUNT/SURE	0.78	0.84
RATIO/ALL	0.82	0.86	RATIO/SURE	0.78	0.85
Success rates on pairs that I expected to be `good`					
RANK/ALL	0.79	0.86	RANK/SURE	0.73	0.81
COUNT/ALL	0.88	0.93	COUNT/SURE	0.83	0.90
RATIO/ALL	0.82	0.86	RATIO/SURE	0.84	0.91
Success rates on pairs that I expected to be `bad`					
RANK/ALL	0.86	0.88	RANK/SURE	0.83	0.90
COUNT/ALL	0.74	0.78	COUNT/SURE	0.71	0.78
RATIO/ALL	0.81	0.85	RATIO/SURE	0.70	0.78
Success rates on pairs that the model predicted to be `good`					
RANK/ALL	0.88	0.86	RATIO/SURE	0.85	0.90
COUNT/ALL	0.82	0.83	COUNT/SURE	0.79	0.82
RATIO/ALL	0.85	0.86	RATIO/SURE	0.79	0.83
Success rates on pairs that the model predicted to be `bad`					
RANK/ALL	0.76	0.85	RANK/SURE	0.70	0.81
COUNT/ALL	0.82	0.91	COUNT/SURE	0.76	0.87
RATIO/ALL	0.78	0.85	RATIO/SURE	0.77	0.89

Table 4.23: Classification success rates of the six models for all test pairs, and for the test pairs I was `sure` about. Success means the `Sense` I *expected* (`good` or `bad`) is the same as the model *predicted*. The best model rates are bolded: RANK/SURE in two kinds of success, RATIO/SURE in three. See Table 4.24 for their success counts. (Note, though, that COUNT/ALL is nearly tied with RATIO/SURE for overall success, and others are not far behind).

Model	On all	On `sure`	Model	On all	On `sure`
Overall success rates					
RANK/ALL	0.63	0.72	RANK/SURE	0.70	0.78
COUNT/ALL	0.72	0.79	COUNT/SURE	0.71	0.76
RATIO/ALL	0.65	0.75	RATIO/SURE	**0.73**	**0.79**
Success rates on pairs that I expected to be `good`					
RANK/ALL	0.67	0.71	RANK/SURE	0.58	0.61
COUNT/ALL	0.84	0.84	COUNT/SURE	0.80	0.77
RATIO/ALL	0.69	0.74	RATIO/SURE	**0.89**	**0.87**
Success rates on pairs that I expected to be `bad`					
RANK/ALL	0.60	0.72	RANK/SURE	**0.80**	**0.92**
COUNT/ALL	0.62	0.75	COUNT/SURE	0.64	0.75
RATIO/ALL	0.62	0.75	RATIO/SURE	0.60	0.72
Success rates on pairs that the model predicted to be `good`					
RANK/ALL	0.58	0.69	RANK/SURE	**0.70**	**0.86**
COUNT/ALL	0.64	0.74	COUNT/SURE	0.64	0.72
RATIO/ALL	0.60	0.72	RATIO/SURE	0.65	0.73
Success rates on pairs that the model predicted to be `bad`					
RANK/ALL	0.69	0.74	RANK/SURE	0.70	0.73
COUNT/ALL	0.83	0.84	COUNT/SURE	0.80	0.79
RATIO/ALL	0.71	0.77	RATIO/SURE	**0.87**	**0.87**

Table 4.24 displays their success counts and rates for the test data, separately for all 100 pairs and for the 67 `sure` pairs, with their winning success rates from Table 4.23 in bold.

The performance of the best models on the anecdotal examples Although RATIO/SURE wins three of the five comparisons, it is the only one of our six models that fails *either* of my primary running examples, and it fails them *both*. I expected `omena`, *apple*, to be similar in meaning to `appelsiini`, *orange*, and RATIO/SURE predicts the pair to be `bad`. I expected `omena` to *not* be similar in meaning to `vero#uudistus`, *tax reform*, and RATIO/SURE predicts the pair to be `good`.

The first failure is clearly a failure of the RATIO/SURE, since it is in the two categories where the model was otherwise the best: expected `good`, and predicted `bad`.

The other winning model, RANK/SURE, was correct for all four of the running examples.

Finally, both RANK/SURE and RATIO/SURE were correct for `omena` and `lanka`, *thread*, which was difficult for other models.

Table 4.24: Success rates for test pairs of the two models that seem best in Table 4.23. Success means that the `Sense` I *expected* (`good` or `bad`) is the same as the model *predicted*. Of the six models, RANK/SURE had the best success rates for the test pairs that I expected to be `bad`, and for the test pairs that it predicted to be `good`; RATIO/SURE had the best success rates for the test pairs that I expected to be `good`, for the test pairs that it predicted to be `bad`, and for the test pairs overall.

RANK/SURE success rates for all 100 test pairs

	Predicted `bad`	Predicted `good`	Rate
Expected `bad`	**44**	11	**0.80**
Expected `good`	19	**26**	0.58
Rate	0.70	**0.70**	

Overall success rate 0.70

RANK/SURE success rates for the 67 `sure` test pairs

	Predicted `bad`	Predicted `good`	Rate
Expected `bad`	**33**	3	**0.92**
Expected `good`	12	**19**	0.61
Rate	0.73	**0.86**	

Overall rate 0.78

RATIO/SURE success rates for all 100 test pairs

	Predicted `bad`	Predicted `good`	Rate
Expected `bad`	**33**	22	0.60
Expected `good`	5	**40**	**0.89**
Rate	**0.87**	0.65	

Overall success rate **0.73**

RATIO/SURE success rates for the 67 `sure` test pairs

	Predicted `bad`	Predicted `good`	Rate
Expected `bad`	**26**	10	0.72
Expected `good`	4	**27**	**0.87**
Rate	**0.87**	0.73	

Overall success rate **0.79**

Chapter 5

Results and further work

Thus far I've presented a conceptual discussion on the distributional and semantic similarity of words. I then provided a sort of mathematical overview of a central component of the computation of distributional similarity, with particular attention to the formula used. Next, a concrete exercise on computing the distributional similarity lists from a corpus was conducted, and examples were shown of the success and failure of a tail word in such a list being semantically similar to the head word. Finally, I developed a new method that could be used to analyse and improve such similarity lists.

The concepts Words are distributionally similar, or contextually similar, if they occur in the same contexts. Distributional similarity is a matter of degree, so it is better to say that words are distributionally similar *to the extent that* they occur in the same contexts.

I operated without an actual definition of semantic similarity, but with the understanding that the *meanings* of semantically similar words are somehow close, core cases being synonymy, close hyponymy, and antonymy. In the end, I resorted to my own intuitive judgments.

When attempting to use distributional similarity as a practically computable substitute for semantic similarity, I can observe only a limited form of context – in my case, single syntactically linked words. In addition, I observe only actual text: not all contexts where a word in some sense *can* occur but only some contexts where it actually *does* occur. Then it is an empirical question concerning how well the substitution works.

My classification of similarity formulas I developed a uniform point of view where the various distributional similarity formulas in use are thought to belong to one of three broad groups, according to how they represent the word and how they deal with the representations of words. All three kinds of

representations assign numeric weights to the attributes of the word. Other sources might call these attributes 'features' or 'properties'. One group of formulas treats the word representations as weighted 'sets' and uses operations analogous to ordinary set intersection and union. This is achieved by generalising the multiset notions of these operations to arbitrary, non-negative weights. Doing this, the weighted Jaccard formula, as used by Grefenstette, has the same form as the ordinary Jaccard.

Another group treats the data as elements of a vector space and uses addition, scaling, and the angle between the vectors. This group (or cosine at least) is important in methods that manipulate a matrix with rows that represent the words. Nonetheless, these methods were beyond the scope of my analysis. Their main idea is the reduction of the dimensionality of the space of the word representations.

The third group of formulas treats the data as discrete densities, also known as probability mass functions. Much of this group builds on the information-theoretic notion of relative entropy. An important case is the information radius. I was fortunate to find an early reference and I extend my thanks to the publisher for putting it on the web. The formula has been rediscovered independently at least twice. It is therefore also known as the Jensen-Shannon divergence and as mean divergence to the mean. The general forms of the information radius and the Jensen-Shannon divergence look different, but they are indeed equivalent.

One of the word-similarity formulas is often presented as the 'block distance' of vectors, even when it is applied to discrete densities, which do not form a vector space. (Several other names are used, including the 'L_1 norm' which seems a misnomer to me when the objects do not belong to a vector space.) In the end, I think this formula belongs among the probability formulas, with the different name *variational distance*, which I also found in the information-theoretic literature: it is the maximal difference of discrete probability *measures*, which has exactly the form of the block distance of vectors when expressed in terms of the corresponding probability *densities*. (Finiteness of the sample space might be essential here.)

Another information-theoretic formula, by Dekang Lin, did not initially fit in my three-way classification of the distributional similarity formulas, because it did not represent words in terms of the weights of the attribute words. I was able to re-express it in a familiar form, but then it appeared to belong to my first group – 'set' formulas – with very specific weights for the attributes. These weights, though they arose from probability assignments, were not in the form of a discrete density for each word. This means that I cannot simply state that the formally identical formula appears in two groups, unlike for the case of variational distance. (Confusion probability

also did not fit in my classification.)

The computation of similarity lists A corpus of Finnish newspaper text was transformed into distributional *similarity lists* for its frequent nouns, leaving all linguistic analysis to a syntactic dependency parser that was available to me at the time. (The collection of all similarity lists was also called a similarity table.) The only use of corpus metadata was to exclude a large class of documents that were not ordinary running text: the television and radio listings.

The 'frequent nouns' were base forms with more than a hundred occurrences labeled `N` by the parser, ignoring other possible labels that remained for the token. These included both common and proper nouns. Each similarity list was a list of one hundred *tail words*, which were those frequent nouns distributionally most similar to a given *head word*, in a decreasing order of their similarity to the head word. Since the head words and tail words of the lists were taken from the same set of distributional word representations, each head word was also usually the first tail word in the list.

The distributional representation of each frequent noun consisted of the numerical weights for the computational *attributes* of the noun. These attributes were the 'major class' words that the dependency parser linked directly to the noun, together with the dependency relation label and an indication of whether the noun was the head or the dependent. In the 'major class' words, I included possible nouns, adjectives and verbs, and exluded other word classes. This choice was somewhat arbitrary. The omission of adverbs is not likely to have had a significant effect, since they are usually not supposed to be linked to nouns.

I included all the attributes that occurred with the word at least once. These can still occur with other words and their cumulative effect on similarity might be positive. On the other hand, the attributes that only occurred once or twice in the whole corpus could have been omitted, because they cannot be shared by different words. Very frequent attributes might also possibly be omitted because they are not likely to be informative. However, I did not omit them in my experiments.

As far as I see, the use of dependency-linked attributes for tasks such as this is still an interesting research topic. My impression is that this method produces better quality than the use of tokens that occur merely somewhere near the word tokens. This impression was partly formed in an earlier, unreported, computation in which I compared syntactic and merely nearby attributes, using the parser to reduce tokens to base forms in both cases. While the parser appears to contain a useful amount of linguistic informa-

tion, I have not yet studied the matter further.

The weights of my attributes were the simple frequencies of co-occurrence with the noun, normalised to the sum of one so that I could use a similarity formula that expects the representations to be probability mass functions. This simple choice fails to account for the different informativeness of the different attributes.

The parser was used for several purposes: to segment the corpus into a stream of tokens, to identify the tokens as the forms of various nouns, adjectives, verbs and so on, and to identify some pairs of such words in the stream as being in specific dependency relations with each other. I used the frequencies of these word-relation-word triples to build the computational representations of thousands of frequent nouns.

Then I used each of these thousands of nouns as a head word of a similarity list that consists of the one-hundred tail nouns whose representations were most similar to the representation of the head word, measured by the information radius, in the order of their decreasing similarity to the head word. This is a usual procedure. An alternative would be to build clusters.

The exploration of the lists I then studied my similarity lists in various ways. I first identified the influential shared pieces of context that made some words relatively similar to each other. I then identified some semantic successes and some failures. A most spectacular failure was caused by a single context item being exceptionally frequent with two different words: the prevalence of the Finnish expressions for *green apple* and *green tax reform* in the corpus made the Finnish words for *apple* and *tax reform* appear distributionally similar. This was partly due to the parser failing to identify many of the occurrences of *green apple* as proper names, and partly due to the accident that an environmentally motivated tax reform was a topic of discussion in the corpus. In any case, this demonstrates that a small number of shared words can have an inordinate effect in the calculation of distributional similarity.

The main observation was that the distributional similarity lists contained both words that were semantically appropriate and words that were semantically inappropriate, and that there were both kinds among the tail words that were distributionally *most* similar to the head word. It is interesting to find ways to identify these two classes of tail words on each list.

Improving the lists Continuing the exploration, I identified a handful of simple numerical characteristics of the pairs of the representations of words. These would be used as further distributional variables in a statistical clas-

sifier, in addition to the information radius that I used as my distributional similarity formula.

The further variables included the number of attributes of both the head word and tail word, the number of attributes they shared, the proportion of the attributes that each shared with the other, and also the position (called rank) of each in the similarity list of the other. Each tail word ranked low on the list of the head word, of course, since I focused on just that part of the similarity list of the head word. The rank of the head word with respect to the tail word varied. (Dekang Lin has used this as a criterion, looking for what he calls 'respective nearest neighbours'.)

Then, I classified a random sample of a few hundred of the tens of thousands of distributionally most similar pairs as semantically good or bad, using only my native speaker intuition. The result is far from perfect but apparently not wholly unusable. Graphs of the distributional variables show extensive overlap for the two semantic classes, but often there is also a difference in the appropriate direction.

I used the semantically annotated sample of pairs to train statistical classification trees that produce approximations to my intuitive semantic classification when given the distributional data about the pairs of words.

The intended use of such trained classifiers is to have them flag particularly good or bad tail words on the similarity lists. One classifier might be good at identifying semantically good tail words, which should be kept. Another might be good at identifying semantically bad tail words, which should be removed. A particularly succesful classifier would be good at both tasks.

The success rates of my classifiers were somewhat promising. All erred in some cases, of course. I could accept the loss of some good tail words, and the bad tail words that remain, or I could merely flag the candidates to help a human who makes the final decisions.

A problem The intuitive classification of pairs of words into semantically good and semantically bad was surprisingly difficult for me. Two crucial pieces of advice helped me through in the end. First, when in doubt, I should record my doubt. Second, I should decide quickly and not look back. This resulted in two classes of both good and bad pairs: those I was sure about, and those I was not sure about. Nevertheless, judging the pairs still often felt unnatural.

The quality of the resulting classes is still poor, or at least doubtful. The whole concept of the loose semantic similarity of arbitrary text words seems suspect. It cannot be rejected outright, because many cases are clear after all. Unclear cases might be improved by some amount of semantic

disambiguation of the words. In addition, the difficulty of the task could be quantified by having several human annotators.

The right method in the future might be to reject the unclear cases from the training data altogether. My doubts notwithstanding, my training set seems to have been usable, and the most useful part appears to be the clear cases, about which I was sure.

Variations on the theme Two questions are specific to the new method of improving the base line similarity lists. First, could better distributional variables be used as input to the classifiers? Second, would some other classification method be better?

Regarding the first question, I had several cumulative sums of weights or proportions of weights in my set of variables at one time. These were omitted because they were rather more complicated than those I presented, and it seemed expedient to learn about simpler variables first. There is also some redundancy in the sets I used: many of my variables chase the idea that the important factor is the number or proportion of shared attributes.

Regarding the second question, there are methods that compete with decision trees. For example, support vector machines have been suggested. This type of method might work better.

Other questions concern the notion of distributional computation in general. for example, I could use a better parser, or I could use a parser better. In addition, I could use metadata.

To use a parser better, instead of using the copula as an attribute, I might follow another link across the copula in search of a more semantically informative attribute. I could also try to omit uninformative attributes altogether, though I did observe a case where even the verb `olla`, *be*, appeared appropriately informative when it was labeled with a locative dependecy relation: it made words for *pocket* and *bag* similar.

The attribute weighting representations are also quite general. For example, instead of using other words from the same text, with or without annotations, the potential translations from a parallel text or from a dictionary could be used. This, I believe, has been suggested to me by Krister Lindén, who used my data set earlier in one of his thesis papers.

Appendix A

Computation formula

A.1 The information radius is the Jensen-Shannon divergence

Robin Sibson (Sibson, 1969) defined *information radius* (of order 1) in a way that seems to match well the description 'mean divergence from the mean'. In contrast to Lin, below, he begins his discussion by introducing a general formula for a weighted mean of any number of probability measures. Here I adopt his definition for the probability mass functions. This requires a step down from a Lebesgue integral as follows.

Sibson builds on the *information gain of order 1*, $I_1(\mu|\nu)$, for two probability measures μ and ν on the measurable subsets of a set X, μ absolutely continuous with ν. For me, the set X is finite, and all its subsets are measurable.

$$I_1(\mu|\nu) = \int_X \log \frac{\mathrm{d}\mu}{\mathrm{d}\nu}\, \mathrm{d}\mu \tag{A.1}$$

Sibson goes on to suggest that by defining $p = \mathrm{d}\mu/\mathrm{d}\lambda$ and $q = \mathrm{d}\nu/\mathrm{d}\lambda$, where λ is yet another probability measure, the more familiar-looking definition is obtained:

$$I_1(\mu|\nu) = \int_X p \log(p/q)\, \mathrm{d}\lambda \tag{A.2}$$

However, for discrete X, it is better to take λ to be the counting measure, $\lambda(E) = |E|$, not a probability measure. Then p and q are the probability mass functions corresponding to μ and ν, and the result is the crucial correspondence desired:

$$I_1(\mu|\nu) = \sum_{x \in X} p(x) \log \frac{p(x)}{q(x)} = D(p\|q) \tag{A.3}$$

Jianhua Lin (1991) proceeds from $D(p\|m) + D(q\|m)$ for two probability mass functions p and q and their equally weighted mean m, re-expresses it in terms of entropy, generalises to arbitrary weights, names the result *Jensen-Shannon divergence*, and then generalises further to any number of probability mass functions.

I define the mean m of the n probability mass functions $p_1, \ldots, p_n : A \to [0\,..\,1]$ and weights w_k for each $a \in A$ by $m(a) = \sum_k w_k p_k(a)$. The definitions for the information radius $R(\ldots)$ and the Jensen-Shannon divergence $J(\ldots)$ are then as follows:

$$\begin{aligned} R\begin{pmatrix} p_1, \ldots, p_n \\ w_1, \ldots, w_n \end{pmatrix} &= \sum_k w_k D(p_k \parallel m) \\ J\begin{pmatrix} p_1, \ldots, p_n \\ w_1, \ldots, w_n \end{pmatrix} &= Hm - \sum_k w_k Hp_k \end{aligned} \tag{A.4}$$

The next step is to prove that the two are one. The proof is a completely straightforward manipulation of the expression for $R(\ldots)$ until the expression for $J(\ldots)$ emerges:

$$\sum_k w_k D(p_k \parallel m) = \sum_k w_k \sum_{a\in A} p_k(a) \log \frac{p_k(a)}{m(a)} \tag{A.5}$$

$$= \sum_k w_k \sum_{a\in A} \big(p_k(a) \log p_k(a) - p_k(a) \log m(a)\big) \tag{A.6}$$

$$= \sum_k w_k \sum_{a\in A} p_k(a) \log p_k(a) - \sum_k w_k \sum_{a\in A} p_k(a) \log m(a) \tag{A.7}$$

$$= -\sum_k w_k Hp_k - \sum_k w_k \sum_{a\in A} p_k(a) \log m(a) \tag{A.8}$$

$$= -\sum_k w_k Hp_k - \sum_{a\in A} \Big(\sum_k w_k p_k(a)\Big) \log m(a) \tag{A.9}$$

$$= -\sum_k w_k Hp_k - \sum_{a\in A} m(a) \log m(a) \tag{A.10}$$

$$= Hm - \sum_k w_k Hp_k \tag{A.11}$$

That is all it takes. Lin could have generalised to any number of arguments and arbitrary weights from the start. I am pleased to have *two* elegant expressions for this important divergence measure.

A.2 The radius from the shared attributes

Dagan, Lee and Pereira state that the equally weighted information radius of two probability mass functions, $R(p, q)$, can be computed from the common points of the two probability distributions $p, q : A \to [0\,..\,1]$. (They call it the Jensen-Shannon divergence and write it as $JS(p, q)$.)

They state that this can be seen by grouping the terms appropriately. I proceed to do just that. The work turns out to consist of bookkeeping.

Auxiliary functions I name the mean: $m = (p+q)/2$. As defined, $m(a) = (p(a) + q(a))/2$.

I use pointwise entropy: $h(x) = -x \log x$; $h(0) = 0$.

I use pointwise radius: $r(x, y) = -\frac{1}{2}(h(x + y) - h(x) - h(y))$.

I name the logarithm of 2, just to keep track of the point where the base of the logarithm matters: $u = \log 2$.

Shorthand sum notation Some shorthand is used, so that instead of $\sum_{a\in M} f(a)$, I write $\sum_M f$. For example:

$$\begin{aligned} \sum\nolimits_A h(p + q) &= \sum_{a\in A} h(p(a) + q(a)) \\ Hp &= \sum\nolimits_A h(p) \end{aligned} \tag{A.12}$$

This is merely shorthand for the duration of this proof.

Partition of the set of attributes I partition A into four named subsets as follows.

$$\begin{aligned} B &= \{a \in A \mid p(a) > 0, q(a) = 0\} \\ C &= \{a \in A \mid p(a) = 0, q(a) > 0\} \\ D &= \{a \in A \mid p(a) > 0, q(a) > 0\} \\ E &= \{a \in A \mid p(a) = 0, q(a) = 0\} \end{aligned} \tag{A.13}$$

Of these, E is uninteresting, since $\sum_E h(p) = 0$ and $\sum_E h(q) = 0$ and so on. I shall also use the following facts:

$$\begin{aligned} \sum\nolimits_A p &= \sum\nolimits_B p + \sum\nolimits_D p \\ \sum\nolimits_A q &= \sum\nolimits_C q + \sum\nolimits_D q \end{aligned} \tag{A.14}$$

Definition I adopt Jianhua Lin's definition, since $R(p,q) = J(p,q)$:

$$R(p,q) = Hm - (Hp + Hq)/2 \tag{A.15}$$

The theorem With the notation now in place, the statement to prove is:

$$R(p,q) = u - \textstyle\sum_D r(p,q) \tag{A.16}$$

A lemma I establish an auxiliary result that disposes of the mean distribution $m = (p+q)/2$. Remember that $u = \log 2$.

$$\begin{aligned}
\textstyle\sum_M h(m) &= \textstyle\sum_M (-\frac{p+q}{2} \log \frac{p+q}{2}) \\
&= \tfrac{1}{2} \textstyle\sum_M (-(p+q))(\log(p+q) - u) \\
&= \tfrac{1}{2} \textstyle\sum_M (-(p+q)\log(p+q) + (p+q)u) \\
&= \tfrac{1}{2} \textstyle\sum_M h(p+q) + \tfrac{u}{2} \sum_M p + \tfrac{u}{2} \sum q
\end{aligned} \tag{A.17}$$

Proof of the theorem The plan now is to partition the three sums that appear when I open the entropies, and to simplify separately the sums over B, C, and D. All sums over E vanish at the outset. The sums over B and C are simple; the sum over D contains the term of interest, and terms that combine nicely with the sums over B and C.

Now I introduce the $\sum$'s hidden in H and rewrite each $\sum_A$.

$$\begin{aligned}
R(p,q) &= \textstyle\sum_A h(m) - \tfrac{1}{2} \sum_A h(p) - \tfrac{1}{2} \sum_A h(q) \\
&= \textstyle\sum_B h(m) + \sum_C h(m) + \sum_D h(m) \\
&\quad - \tfrac{1}{2}(\textstyle\sum_B h(p) + \sum_D h(p)) \\
&\quad - \tfrac{1}{2}(\textstyle\sum_C h(q) + \sum_D h(q))
\end{aligned} \tag{A.18}$$

Then I regroup the terms and name the three partial sums:

$$\begin{aligned}
S_B &= \textstyle\sum_B h(m) - \tfrac{1}{2} \sum_B h(p) \\
S_C &= \textstyle\sum_C h(m) - \tfrac{1}{2} \sum_C h(q) \\
S_D &= \textstyle\sum_D h(m) - \tfrac{1}{2} \sum_D h(p) - \tfrac{1}{2} \sum_D h(q) \\
R(p,q) &= S_B + S_C + S_D
\end{aligned} \tag{A.19}$$

To simplify S_B, I use my lemma to take $h(m)$ apart. From $q_a = 0$ for $a \in B$, I have that $\sum_B h(p+q) = \sum_B h(p)$, so two of the terms cancel, and the term involving $\sum_B q$ vanishes.

$$\begin{aligned} S_B &= \tfrac{1}{2}\sum\nolimits_B h(p+q) + \tfrac{u}{2}\sum\nolimits_B p + \tfrac{u}{2}\sum\nolimits_B q - \tfrac{1}{2}\sum\nolimits_B h(p) \\ &= \tfrac{u}{2}\sum\nolimits_B p \end{aligned} \tag{A.20}$$

Similarly, to simplify S_C, I use my lemma to take $h(m)$ apart. From $p_a = 0$ for $a \in C$, I have that $\sum_C h(p+q) = \sum_C h(q)$, so again two of the terms cancel, and the term involving $\sum_C p$ vanishes.

$$\begin{aligned} S_C &= \tfrac{1}{2}\sum\nolimits_C h(p+q) + \tfrac{u}{2}\sum\nolimits_C p + \tfrac{u}{2}\sum\nolimits_C q - \tfrac{1}{2}\sum\nolimits_C h(p) \\ &= \tfrac{u}{2}\sum\nolimits_C q \end{aligned} \tag{A.21}$$

To simplify the most interesting term S_D, I again use my lemma to take $h(m)$ apart. None of the terms cancel or vanish, but three of them join to become $\sum_D r(p,q)$. The remaining two will combine with S_B and S_C.

$$\begin{aligned} S_D &= \tfrac{1}{2}\sum\nolimits_D h(p+q) + \tfrac{u}{2}\sum\nolimits_D p + \tfrac{u}{2}\sum\nolimits_D q - \tfrac{1}{2}\sum\nolimits_D h(p) - \tfrac{1}{2}\sum\nolimits_D h(q) \\ &= \tfrac{u}{2}\sum\nolimits_D p + \tfrac{u}{2}\sum\nolimits_D q - \sum\nolimits_D r(p,q) \end{aligned} \tag{A.22}$$

Combining everything in one equation again and noticing that $\sum_A p = 1$ and $\sum_A q = 1$, the following is the result.

$$\begin{aligned} R(p,q) &= \tfrac{u}{2}\sum\nolimits_B p + \tfrac{u}{2}\sum\nolimits_C q + \tfrac{u}{2}\sum\nolimits_D p + \tfrac{u}{2}\sum\nolimits_D q - \sum\nolimits_D r(p,q) \\ &= \tfrac{u}{2}\left(\sum\nolimits_A p + \sum\nolimits_A q\right) - \sum\nolimits_D r(p,q) \\ &= u - \sum\nolimits_D r(p,q) \end{aligned} \tag{A.23}$$

This was to be proved.

Comment The result was indeed obtained by grouping the terms appropriately. No difficulties were encountered apart from the bookkeeping.

A.3 The pointwise radius is never negative

The pointwise radius $r(x,y)$ is never negative for probabilities x and y. This is simple to prove. First, for either argument zero, $r(x,y)$ is zero:

$$\begin{aligned} r(0,y) &= -\tfrac{1}{2}(h(y) - h(0) - h(y)) = -\tfrac{1}{2}h(0) = 0\ ; \\ r(x,0) &= -\tfrac{1}{2}(h(x) - h(x) - h(0)) = -\tfrac{1}{2}h(0) = 0\ . \end{aligned} \tag{A.24}$$

Second, for both $x > 0$ and $y > 0$, the expression can be manipulated simply to reveal its positive nature:

$$\begin{aligned} 2r(x,y) &= -(h(x+y) - h(x) - h(y)) \\ &= (x+y)\log(x+y) - x\log x - y\log y \\ &= x(\log(x+y) - \log x) + y(\log(x+y) - \log y) \\ &= x\log\frac{x+y}{x} + y\log\frac{x+y}{y} \\ &= x\log(1+y/x) + y\log(1+x/y) > 0\ . \end{aligned} \tag{A.25}$$

The last inequality follows from the facts that the sums, products and ratios of positive numbers are positive, the sum of two positive numbers is greater than the first number, and the logarithm of a number is positive when the number is greater than 1. All the logarithms above are defined because x and y are positive in this branch of the proof. Finally, from $2r(x,y) > 0$, it is only a very short step to $r(x,y) > 0$.

(But I think it should be possible to see the result as a direct consequence of the convexity of $x\log x$.)

Appendix B

My semantic judgments on the training and test pairs

The eight tables that follow are complete listings of the four classes of head–tail word pairs in the training and test samples that I made for my experiment. First, B.1 are those training pairs that I judged to be good and was sure about it. Second, B.2 are those training pairs that I also judged to be good but was not sure about it. Together these tables contain all the training pairs that I judged to be good. Third, B.3 are those training pairs that I judged to be bad but was not sure about it. Fourth, B.4 are those training pairs that I judged to be bad and was sure about it. Together these last two tables contain all pairs that I judged to be bad. The first and fourth table contain all the training pairs that I was sure about, and the second and third table contain those that I was not sure about.

Tables B.5 to B.8 give the corresponding classification of the 100 test pairs.

Table B.1: Good and sure (140 training pairs)

Head	Tail
-#teos	luettelo
12#-	13#-
1300-luku	1600-luku
1950-luku	50-luku
2000-luku	90-luku
ahdistelu	perhe#väki#valta
aikuis#koulutus#keskus	taide#museo
ajo#kielto	rangaistus
ajo#rata	silta
ali#hankkija	asiakas
ali#jäämä	loppu#summa
ammattilais#turnaus	turnaus
analysointi	seuranta
asunto#tuotanto	tuotanto
bio#jäte	romu
budjetti#ali#jäämä	velka
dollari	frangi
edustaja	johtaja
epäilys	käsitys
eteneminen	valmistelu
eurocup	alku#lohko
hasis	marihuana
henkilöstö#meno	maksu#tulo
hertta	ruutu#ässä
humanismi	kulttuuri
idoli	suosikki
ilmi#anto	kantelu
ilo	tunne
iso#äiti	lapsi
itse#varmuus	kilpailu#etu
itsenäisyys	yhtenäisyys
joulu#aatto	perjantai-#ilta
jousi#kvartetto	sello#konsertto
jälleen#rakennus	sota
järistys	törmäys
jäsenistö	tuomaristo
keilaaja	heittäjä
keitto#kirja	oppi#kirja

Table B.1: Good and sure (training, continued)

Head	Tail
kellari#kerros	ala-#aula
keskustelu#tilaisuus	muisto#tilaisuus
kieli#taido	osaaminen
kirja#messu	konferenssi
kirkko#neuvosto	vesi#oikeus
koitos	peli
kokaiini	heroiini
kolmas#osa	neljäs#osa
kolmois#voitto	kaksois#voitto
kompromissi	päätös
koppari	pelaaja
korko#markkina	markkina
koulutus#tuki	startti#raha
kulta#mitali	olympia#mitali
kulttuuri#piiri	kulttuuri#elämä
kunta-#ala	vakuutus#ala
kuponki	äänestys#lippu
kuva#kirja	opas
kymmenys	sadas#osa
kävely#matka	väli#matka
lahjoitus	bonus
laulu#juhla	juhla#viikko
lehti#kirjoitus	lausunto
leikkaus	alennus
leivonnainen	kakku
levytys	teos
liikunta#toimi	nuoriso#asiain#keskus
liioittelu	yli#lyönti
lisä#osa	työttömyys#päivä#raha
lisä#tulo	lisä#tuki
lisääminen	lisääntyminen
luokan#opettaja	insinööri
luopuminen	kieltäytyminen
maailman#luokka	koko#luokka
maasto#hiihto	mäki#hyppy
maestro	muusikko
markkina#talous	politiikka
mekko	asu

Table B.1: Good and sure (training, continued)

Head	Tail
meri#alue	saaristo
mestaruus#kisa	olympia#karsinta
miekka	puukko
nousija	ennakko#suosikki
oikeus#käytäntö	laki
olo#huone	työ#huone
oma#elämäkerta	muistelma#teos
opetus#neuvos	lähetystö#neuvos
opinto#raha	toimeentulo#tuki
oppositio#puolue	hallitus#puolue
paperi#liitto	pankki#toimi#henkilö#liitto
parlamentti#vaali	kunnallis#vaali
prostituutio	tupakointi
puhelin#yhtiö	yritys
puu#tavara	vilja
pyrbasket	ilves
päivä#hoito#maksu	hoito#raha
pää#juhla	konsertti
päättäjä	luottamus#henkilö
pääty	seinä
rauhan#turva#operaatio	projekti
seutu#kaava	direktiivi
solidaarisuus	yhteis#työ
startti	lähtö
strategia	ohjelma
suunnittelu#kilpailu	tarjous#kilpailu
suur#lähetystö	edustusto
suvaitsevaisuus	avoimuus
sähkö#markkina	pääoma#markkina
säiliö#auto	kaivin#kone
tanssi#musiikki	tango
tapa#kulttuuri	perinne
tarkastaja	viran#omainen
tasoitus#maali	avaus#maali
tekniikka	menetelmä
teko#järvi	pato
tele#visio	tiedotus#väline
timjami	leivin#jauhe

Table B.1: Good and sure (training, continued)

Head	Tail
toimi#ala#johtaja	projekti#päällikkö
torstai	tiistai
touko-#kesäkuu	vuoden#vaihde
tshaikovski	rahmaninov
tulo#loukku	loukku
tutkimus#johtaja	talous#johtaja
tuuli	luoteis#tuuli
tyhmyys	tietämättömyys
työ#määrä	työ#taakka
työ#paikka	koulutus#paikka
työ#valio#kunta	edus#kunta#ryhmä
ulko#politiikka	energia#politiikka
vara#mies	apulainen
vara#puheen#johtaja	ryhmän#johtaja
varuste	työ#väline
vasten#mielisyys	paheksunta
veiterä	honka
vero#vähennys	korko#tuki
veto-#oikeus	harkinta#valta
vihannes	tomaatti
yhden#vertaisuus	oikeus#turva
yksityis#henkilö	yrittäjä
yli#oppilas#tutkinto	korkea#koulu#tutkinto
ylä#puoli	ulko#puoli
ympäristö#lauta#kunta	sosiaali#lauta#kunta
yrittäminen	työn#teko

Table B.2: Good but unsure (88 training pairs)

Head	Tail
alas#tulo	slammi
ammattilainen	henkilö
arvio	ilmoitus
esitys#lista	suunnite
fyysikko	säveltäjä
hinta#taso	työttömyys#prosentti
home	tulva
huippu#hetki	yllätys
huippu#pelaaja	osan#ottaja
ihmis#arvo	oikeus#turva
isku#ryhmä	jury
joukkue#kilpa	vieras#peli
kansalaisuus	nato-#jäsenyys
kauppa#halli	yli#oppilas#talo
kaupungin#museo	kaupungin#valtuusto
keho	aivo
keittiö#mestari	liike#mies
kiinteistön#välittäjä	rakennus#insinööri
kilpaileminen	soittaminen
kirja#kauppa	toimisto
kokonais#määrä	markkina-#arvo
konkari	kaveri
korko#markkina	sisä#politiikka
koti#markkina	vienti#teollisuus
kumppanuus	vapaa#ehtoisuus
laskema	jakama
laskettelu#rinne	mökki
linja	järjestelmä
loppiainen	lauantai
loppu#osa	lähtö#hinta
lupa#ehto	perustus#laki
maltti	asian#tuntemus
massa	kokonaisuus
massa	joukko
moni#muotoisuus	eko#systeemi
monopoli#asema	etu#sija
muistiin#pano	kantelu
murros	risti#riita

Table B.2: Good but unsure (training, continued)

Head	Tail
naapuri#talo	kylä
neuvottelija	päättäjä
opettajan#koulutus	kieli#kylpy
osake#salkku	velka#taakka
osallistumis#maksu	neliö#hinta
osasto	pankki
palvelu#työn#antaja	työn#antaja#liitto
parlamentaarikko	asian#tuntija
peluu	kirjoittaminen
poikkeus	pää#asia
presidentti#ehdokas	ryhmän#johtaja
pujottelu	kokonais#kilpailu
päivä#määrä	tieto
päätöksen#teko#järjestelmä	lain#säädäntö
rock	urheilu
ruis#leipä	suklaa
rynnistys	taka#isku
sala#liitto	ryhmä
slu	työ#ryhmä
sosiaali#turva	lain#säädäntö
sosiaali#virasto	rakennus#lauta#kunta
sota#invalidi	kansalainen
sovinto#ehdotus	rikos#ilmoitus
suhde	rooli
taistelija	hyppääjä
tapaus	asia
tarha	maa
tasku	piilo
tematiikka	perus#kysymys
terveyden#tila	työ#tilanne
toissa#vuosi	kiirastorstai
tunnustus#palkinto	määrä#raha
uutinen	esi#merkki
vaellus	tapahtuma
valjakko	ponseri
valo	tila
valtio	kansalainen
valvoja	järjestäjä

Table B.2: Good but unsure (training, continued)

Head	Tail
vapaa#ohjelma	finaali
vapaa-#aika	liikunta
vara#puhe#mies	perustus#laki#valio#kunta
vauhti	kyyti
velka#kirja	omaisuus
veron#korotus	pudotus
veto#apu	vahvistus
viivästys#korko	päivä#hoito#maksu
vranitzky	simitis
vuokra#tulo	tienesti
yhtiö#kokous	istunto
ylilääkäri	ministeri

Table B.3: Bad but unsure (47 training pairs)

Head	Tail
ala#otsikko	perus#ajatus
ashkenazy	ay-liike
ashkenazy	peterka
asunto#markkina	hinta#taso
asunto#tuotanto	väki#luku
auttama	neuvottelema
café	-#tapahtuma
elämän#muoto	systeemi
evergreen	empress
fenno	rahasto
höyhen	ruusu
keitto#kirja	käännös
komeus	ikä
koriste	esi#kuva
koti#apulainen	asiakas
kulttuuri#elämä	pankki#järjestelmä
kuri	pihti
laji#tyyppi	systeemi
lataus	motivaatio
latina	viulun#soitto
metafora	lyhenne
mihailov	vakkila
ohjaaja	lääkäri
ormo	takko
paneeli	mielen#osoitus
paneeli#keskustelu	arkkitehti#kilpailu
perus#palvelu	hyvin#vointi
perus#rakenne	perus#asia
pohjan#tähti	alminsali
posti#toimi#paikka	porla
produktio	laji
puhe	päätös
puutarhuri	esiintyjä
rahan#jako	lain#säädäntö
ralli#autoilu	a-poiki
rang	jets
rasitus	pulma
rattle	ay-liike

Table B.3: Bad but unsure (training, continued)

Head	**Tail**
riski	määrä
satelliitti	systeemi
sikari	ruoka
storgårds	ekki
säveltäja	valmistaja
tela	uima#liitto
tutkimus#yksikkö	konttori
voima#vara	hyöty
yritys#maailma	markkina

Table B.4: Bad and sure (125 training pairs)

Head	Tail
alusta	maku#asia
aniche	vähi
antaminen	ajaminen
apostoli	alus
asu	laite
avain#henkilö	tunnettuja
avo#meri	vastaisuus
eläin#laji	valinta#peruste
eläin#museo	kansan#talous#tiede
emu-#jäsenyys	sääst
ensi	osa#syy
erityis#asema	sopu#sointu
etelä#kaakko	valta#meri
euro	maa#kaasu
for	balansor
happi	itse#luottamus
henki	asia
herkkyys	ratkaisu
hertta#ässä	varas#lähtö
hevonen	asiakas
hinta#vertailu	kyse
hitunen	disk#ontto#korko
huippu#taso	kolkka
huippu#yksikkö	seuraaja
ihmis#oikeus#sopimus	jalka#pallo#liitto
ikä#raja	hinta#taso
illuusio	kysymys#merkki
inflaatio#vauhti	yö#lämpö#tila
itä#rannikko	mestaruus#kisa
johto#päätös	temppu
johto#tähti	pakko
kaappaaja	pakko
kanto	saalis
kartelli	jolla
kattaus	päivä#lämpö#tila
kehto	vasta#väittäjä
kerrostuma	tieteen#ala
kompostointi	imetys

Table B.4: Bad and sure (training, continued)

Head	Tail
komppania	alus
kärki#joukko	kyyti
käyttäytyminen	ongelma
lasten#tarha	firma
liito-#orava	työttömyys#prosentti
lima#kalvo	vire
logo	alue#valtaus
lokki	metsä#palo#varoitus
lonkero	rokote
lounais#osa	avain#asema
läpi#leikkaus	perus#opetus
maailman#lista	tukko
maan#mies	asiakas
maku#asia	metsä#palo#varoitus
markkina-#alue	huolen#aihe
matka	ura
middlesbrough	pakko
mielekkyys	perus#asia
miljonääri	suosittuja
muslimi	asiakas
myyrmäkitalo	suunnite
noutaja	sade
nürburgring	jolla
olympia#kivääri	trap
omistus#pohja	gsm-#verkko
osa#puu	vaja
otto	jolla
paatos	murros
pakolais#järjestö	perus#kirja
palata	kukka
pantti#vanki	riski
pelle	umpi#kuja
perhos#uinti	ponnahdus#lauta
piina	toimi#kausi
pitsi	parta
polku#pyöräilijä	kaupunki#suunnittelu#lauta#kunta
puheen#johtajuus	jäsen#määrä
päivystys	kyse

Table B.4: Bad and sure (training, continued)

Head	Tail
päivä#lämpö#tila	jolla
raaka	osa#syy
ramppi	asunto
ratti	avain#asema
realisti	loppu#ilta#päivä
rypäle	syy
sankar	asia
sankaritar	vasta#väittäjä
sato	pelaaja
sisä#ilma	nolla
skaala	maksimi#lämpö#tila
spiri	päivä#työ
säde	tunti#vauhti
sääli	op-pirkka
taka#raja	pää#asia
talous#alue	kiertue
talvi#olympialainen	luonto#kuva
tasaisuus	suosittuja
tavata	sanan#valta
telkkari	ikkuna
tilaus#kanta	uskominen
toiminta#ympäristö	systeemi
tuomi	menestyminen
ulko#puoli	puuttuma
ulko#raja	kilpailu#viran#omainen
ulkoilu#maja	määrän#pää
ulottuvuus	ratkaisu
urheilu#laji	tuote#ryhmä
urheilu-#uutinen	suku#kokous
urheilu-#uutinen	-#tilanne
uumen	itä#osa
vaaka#lauta	näkö#piiri
varasto#tila	yllätys
varis	pakko
varmuus	vauhti
vasta#kaiku	super#bingo
vasta#väittäjä	tarkoitus
vero#raha	heijastin

Table B.4: Bad and sure (training, continued)

Head	Tail
viides#osa	maksimi#lämpö#tila
viiva	vire
virkistys#alue	rakennus
voltti	työ#päivä
vähäisyys	loppu#tulos
vähättely	loppu#tulos
väli#aika	tulos
välähdys	varoitus#aika
väri#kuva	summa
yhdys#kunta#palvelu	työttömyys#aste
äänestys#paikka	tapaus

Table B.5: Good and sure (31 test pairs)

Head	Tail
aula	huone
auto-#onnettomuus	rytäkkä
ennustus	ennuste
hyppy	juoksu
jatkuminen	sujuvuus
kauppa-_ja_teollisuusministeriö	sosiaali#ministeriö
konferenssi	kilpailu
kultti	puolue
kunnossa#pito	suojelu
kurssi#nousu	pudotus
melodia	musiikki
muovi#kassi	ämpäri
niska	jalka
olympia#kivääri	keihään#heitto
ongelma	haaste
palkka#luokka	palkka#taso
perustaminen	sulkeminen
pesti	toimi#kausi
poliisi#kuulustelu	kuulustelu
puhelin#yhtiö	yhtiö
rintama	leiri
sekaannus	kohu

Table B.5: Good and sure (test, continued)

Head	Tail
sika	possu
suunnitelma	ajatus
tennis	keilailu
umpi#kuja	kriisi
univormu	paita
vastike	sähkö#lasku
vyöry	myllerrys
yli#lyönti	väärin#käsitys
ystävyys	suhde

Table B.6: Good but unsure (14 test pairs)

Head	Tail
anteeksipyyntö	kommentti
argumentti	todiste
asennus	kokeilu
elohopea	öljy
erikois#joukko	ase#voima
jatko-#opinto	siviili#palvelus
katolilainen	demokraatti
lisensiaatti	arkkitehti
rakentaja	poliitikko
rannikko#seutu	vaali#piiri
tulo#loukku	sääntely
täys#osuma	saalis
vara#rehtori	tiede#kunta
yritys#toiminta	yritys

Table B.7: Bad but unsure (19 test pairs)

Head	Tail
ahtaus	työ#voima#pula
arvo#sana	tuuri
asema#tunneli	keskus#vankila
avaus#kilpailu	loppu#piste
dokumentti#sarja	jalka#palloilu
entsyymi	hyödyke

Table B.7: (Bad but unsure (test, continued)

Head	Tail
humanisti	pelaaja
kasvu#kausi	työ#päivä
kattavuus	yleis#sitovuus
keittiö#mestari	jalka#palloilija
pankki#toiminta	sponsorointi
siirto#summa	tunti#palkka
taito	mahdollisuus
turva#verkko	tie#yhteys
työ#aika	järjestelmä
vanheneminen	ilmaston#muutos
veri#näyte	henkilö#tieto
yli#oppilas	osakas
yritys#osto	kokeilu

Table B.8: Bad and sure (36 test pairs)

Head	Tail
aita	ej
apul	sovinto#ehdotus
etsijä	itsestäänselvyys
fiorenti	vasta#väittäjä
haamu	taso#ero
hartia	eläke
helvetti	kysymys
herätys	op-pirkka
kaivin#kone	uskominen
kellari#teatteri	pää#posti
kikka	lähtö#kohta
komeetta	ryöstäjä
ky	käytäntö
linjata	kruunun#prinssi
lipun#myynti	yli#oppilas#kirjoitus
loma#matka	murto
luku#taito	fiilis
luonnon#tila	katti
lähi#vuosi	vakio
maali#tilanne	työ#paikka

Table B.8: Bad and sure (test, continued)

Head	Tail
maksimi#lämpö#tila	riski-syp
odottama	vaihtama
osallistua	yksityis#näyttely
pelata	telakka#tuki
pelottelu	sinä
risu	tolkku
ruutu	peli
saareke	vaihto#ehto
saatana	osakas
studio	vuoro
tanssima	harjoittelema
tavata	kolmi#vuotis#kausi
tunnettuja	seitsemän#tenä
valmius#joukko	suojelu#alue
veri	kone
äänestys#vilkkaus	päivä#lämpö#tila

Appendix C

The classification trees

The following few pages contain the textual representations of the classification trees, as output by the `rpart` library of R. Below are a handful of typical R expressions and commands that I used, with `data` as my main data frame.

1. `model <- Sense ~ Sim + Rank + ... + PTail`
2. `attach(data)`
3. `fit <- rpart(model)`
4. `logNShared <- log(1 + NShared)`
5. `HShared <- NShared / NHead`
6. `easy <- data[Sense == "sure"]`, with `HShared` and `TShared` merged in `data`.
7. `fit <- rpart(model, easy)`
8. `expected <- Sense[Sense == "bad"]`
9. `predicted <- predict(fit, type = "class")[Sense == "bad"]`
10. `sum(expected == predicted)`

Model with all variables, trained on all 400 pairs This is a textual representation of the classification tree of Figure 4.8 on page 141.

```
n= 400

node), split, n, loss, yval, (yprob)
      * denotes terminal node

  1) root 400 172 good (0.43000000 0.57000000)
    2) NShared< 14.5 117  26 bad (0.77777778 0.22222222) *
    3) NShared>=14.5 283  81 good (0.28621908 0.71378092)
      6) Knar>=70.5 156  69 good (0.44230769 0.55769231)
       12) PHead< 0.5002834 127  62 bad (0.51181102 0.48818898)
         24) Rank< 6.5 18   3 bad (0.83333333 0.16666667) *
         25) Rank>=6.5 109  50 good (0.45871560 0.54128440)
           50) NTail>=465.5 84  40 bad (0.52380952 0.47619048)
            100) NShared< 26.5 16   2 bad (0.87500000 0.12500000) *
            101) NShared>=26.5 68  30 good (0.44117647 0.55882353)
              202) PTail>=0.1841743 60  30 bad (0.50000000 0.50000000)
                404) NShared< 56 28  10 bad (0.64285714 0.35714286) *
                405) NShared>=56 32  12 good (0.37500000 0.62500000)
                  810) Sim< 0.7666014 17   7 bad (0.58823529 0.41176471) *
                  811) Sim>=0.7666014 15   2 good (0.13333333 0.86666667) *
              203) PTail< 0.1841743 8   0 good (0.00000000 1.00000000) *
           51) NTail< 465.5 25   6 good (0.24000000 0.76000000) *
       13) PHead>=0.5002834 29   4 good (0.13793103 0.86206897) *
      7) Knar< 70.5 127  12 good (0.09448819 0.90551181) *
```

Model with counting variables, trained on all 400 pairs This is a textual representation of the classification tree of Figure 4.10 on page 145.

```
n= 400

node), split, n, loss, yval, (yprob)
      * denotes terminal node

  1) root 400 172 good (0.4300000 0.5700000)
    2) NShared< 14.5 117  26 bad (0.7777778 0.2222222)
      4) NShared< 7.5 58   6 bad (0.8965517 0.1034483) *
      5) NShared>=7.5 59  20 bad (0.6610169 0.3389831)
       10) NTail>=135.5 26   4 bad (0.8461538 0.1538462) *
       11) NTail< 135.5 33  16 bad (0.5151515 0.4848485)
         22) PTail>=0.3100244 19   6 bad (0.6842105 0.3157895) *
         23) PTail< 0.3100244 14   4 good (0.2857143 0.7142857) *
    3) NShared>=14.5 283  81 good (0.2862191 0.7137809)
      6) NTail>=465.5 164  62 good (0.3780488 0.6219512)
       12) NShared< 27 18   2 bad (0.8888889 0.1111111) *
       13) NShared>=27 146  46 good (0.3150685 0.6849315)
         26) PHead< 0.4975051 99  40 good (0.4040404 0.5959596)
           52) NTail>=3039.5 20   7 bad (0.6500000 0.3500000) *
           53) NTail< 3039.5 79  27 good (0.3417722 0.6582278)
            106) NShared< 72.5 51  23 good (0.4509804 0.5490196)
              212) PTail>=0.1841743 44  21 bad (0.5227273 0.4772727)
                424) NShared>=39.5 29  11 bad (0.6206897 0.3793103)
                  848) PHead< 0.3189811 13   2 bad (0.8461538 0.1538462) *
                  849) PHead>=0.3189811 16   7 good (0.4375000 0.5625000) *
                425) NShared< 39.5 15   5 good (0.3333333 0.6666667) *
              213) PTail< 0.1841743 7   0 good (0.0000000 1.0000000) *
            107) NShared>=72.5 28   4 good (0.1428571 0.8571429) *
         27) PHead>=0.4975051 47   6 good (0.1276596 0.8723404) *
      7) NTail< 465.5 119  19 good (0.1596639 0.8403361) *
```

Model with ratio variables, trained on all 400 pairs This is a textual representation of the classification tree of Figure 4.11 on page 146.

```
n= 400

node), split, n, loss, yval, (yprob)
      * denotes terminal node

  1) root 400 172 good (0.43000000 0.57000000)
    2) NShared< 14.5 117  26 bad (0.77777778 0.22222222) *
    3) NShared>=14.5 283  81 good (0.28621908 0.71378092)
      6) TShared< 0.09172571 155  63 good (0.40645161 0.59354839)
       12) HShared< 0.3938645 116  56 bad (0.51724138 0.48275862)
         24) TShared< 0.05271287 59  19 bad (0.67796610 0.32203390)
           48) PHead< 0.3765965 38   8 bad (0.78947368 0.21052632) *
           49) PHead>=0.3765965 21  10 good (0.47619048 0.52380952)
             98) TShared< 0.02494479 8   2 bad (0.75000000 0.25000000) *
             99) TShared>=0.02494479 13   4 good (0.30769231 0.69230769) *
         25) TShared>=0.05271287 57  20 good (0.35087719 0.64912281)
           50) TShared>=0.0692699 27  12 bad (0.55555556 0.44444444)
            100) TShared>=0.08840272 7   1 bad (0.85714286 0.14285714) *
            101) TShared< 0.08840272 20   9 good (0.45000000 0.55000000)
              202) PTail>=0.3061464 10   3 bad (0.70000000 0.30000000) *
              203) PTail< 0.3061464 10   2 good (0.20000000 0.80000000) *
           51) TShared< 0.0692699 30   5 good (0.16666667 0.83333333) *
       13) HShared>=0.3938645 39   3 good (0.07692308 0.92307692) *
      7) TShared>=0.09172571 128  18 good (0.14062500 0.85937500) *
```

Model with all variables, trained on 256 sure pairs This is a textual representation of the classification tree of Figure 4.12 on page 152.

```
n= 265

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 265 125 good (0.47169811 0.52830189)
   2) NShared< 14.5 85  12 bad (0.85882353 0.14117647) *
   3) NShared>=14.5 180  52 good (0.28888889 0.71111111)
     6) Knar>=68.5 90  45 bad (0.50000000 0.50000000)
      12) PTail>=0.2722674 31   7 bad (0.77419355 0.22580645) *
      13) PTail< 0.2722674 59  21 good (0.35593220 0.64406780)
        26) NShared< 25 14   4 bad (0.71428571 0.28571429) *
        27) NShared>=25 45  11 good (0.24444444 0.75555556)
          54) Rank< 6.5 8   3 bad (0.62500000 0.37500000) *
          55) Rank>=6.5 37   6 good (0.16216216 0.83783784) *
     7) Knar< 68.5 90   7 good (0.07777778 0.92222222) *
```

Model with counting variables, trained on 256 sure pairs This is a textual representation of the classification tree of Figure 4.13 on page 153.

```
n= 265

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 265 125 good (0.4716981 0.5283019)
   2) NShared< 14.5 85  12 bad (0.8588235 0.1411765) *
   3) NShared>=14.5 180  52 good (0.2888889 0.7111111)
     6) NTail>=465.5 95  39 good (0.4105263 0.5894737)
      12) NShared< 28.5 12   0 bad (1.0000000 0.0000000) *
      13) NShared>=28.5 83  27 good (0.3253012 0.6746988)
        26) PHead< 0.4975051 52  23 good (0.4423077 0.5576923)
          52) NTail>=2666.5 14   2 bad (0.8571429 0.1428571) *
          53) NTail< 2666.5 38  11 good (0.2894737 0.7105263) *
        27) PHead>=0.4975051 31   4 good (0.1290323 0.8709677) *
     7) NTail< 465.5 85  13 good (0.1529412 0.8470588) *
```

Model with ratio variables, trained on 256 sure pairs This is a textual representation of the classification tree of Figure 4.14 on page 154.

```
n= 265

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 265 125 good (0.47169811 0.52830189)
   2) NShared< 14.5 85  12 bad (0.85882353 0.14117647)
     4) PTail>=0.3087822 57   3 bad (0.94736842 0.05263158) *
     5) PTail< 0.3087822 28   9 bad (0.67857143 0.32142857)
      10) TShared< 0.06019476 15   1 bad (0.93333333 0.06666667) *
      11) TShared>=0.06019476 13   5 good (0.38461538 0.61538462) *
   3) NShared>=14.5 180  52 good (0.28888889 0.71111111)
     6) TShared< 0.04772655 52  25 bad (0.51923077 0.48076923)
      12) HShared< 0.4167826 35   9 bad (0.74285714 0.25714286)
        24) PTail>=0.1885482 26   3 bad (0.88461538 0.11538462) *
        25) PTail< 0.1885482 9   3 good (0.33333333 0.66666667) *
      13) HShared>=0.4167826 17   1 good (0.05882353 0.94117647) *
     7) TShared>=0.04772655 128  25 good (0.19531250 0.80468750)
      14) HShared< 0.08851655 12   5 bad (0.58333333 0.41666667) *
      15) HShared>=0.08851655 116  18 good (0.15517241 0.84482759) *
```

Bibliography

R. H. Baayen, 2008. *Analyzing linguistic data: A practical introduction to statistics using R.* Cambridge University Press. [59]

James O. Berger, 1985. *Statistical Decision Theory and Bayesian Analysis.* Springer Series in Statistics. Second edition. [46]

Max Bramer, 2007. *Principles of Data Mining.* Undergraduate Topics in Computer Science. Springer. [-]

Eugene Charniak, 1993. *Statistical Language Learning.* The MIT Press. [37]

Kenneth Ward Church and Patrick Hanks, 1989. Word association norms, mutual information, and lexicography. In *27th Annual Meeting of the ACL*, pages 76–83. [45]

Kenneth Ward Church and Patrick Hanks, 1990. Word association norms, mutual information, and lexicography. *Computational Linguistics*, 16:22–29. [45]

Thomas M. Cover and Joy A. Thomas, 1991. *Elements of Information Theory.* Wiley. [42, 43, 45, 46]

James Richard Curran, 2003. *From Distributional to Semantic Similarity.* Ph.D. thesis, University of Edinburgh. [24, 35, 38]

Ido Dagan, Lillian Lee, and Fernando Pereira, 1997. Similarity-based methods for word sense disambiguation. In *Proceedings of ACL-EACL '97*, pages 56–63. [46, 48]

Ido Dagan, Lillian Lee, and Fernando C. N. Pereira, 1999. Similarity-based models of word cooccurrence probabilities. *Machine Learning*, 34(1–3):43–69. Special issue on natural language learning. [46]

Ido Dagan, Shaul Marcus, and Shaul Markovitch, 1995. Contextual word similarity and estimation from sparse data. *Computer Speech and Language*, 9:123–152. [11, 24]

Ute Essen and Volker Steinbiss, 1992. Cooccurrence smoothing for stochastic language modeling. In *Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, volume I, pages 161–164. [36, 43]

Christiane Fellbaum, editor, 1998. *WordNet: an electronic lexical database*. The MIT Press. [27]

J. R. Firth, 1957. A synopsis of linguistic theory 1930–55. In *Studies in linguistic analysis*, pages 1–32. Oxford. Reprinted in Palmer (1968), chapter 11, pages 168–205. [21]

Paul L. Garvin, 1962. Computer participation in linguistic research. *Language*, 38(4). [25]

I. J. Good, 1977. Explicativity: a mathematical theory of explanation with statistical applications. *Proceedings of the Royal Society of London, Series A*, 354:303–330. Full text is available from royalsocietypublishing.org, doi: 10.1098/rspa.1977.0069, and an abridged republication as chapter 23 of (Good, 1983). [46]

I. J. Good, 1979. A. M. Turing's statistical work in World War II. *Biometrika*, 66(2):393–396. This appears to be number XXXVII in a series titled Studies in the History of Probability and Statistics. [46]

Irving J. Good, 1983. *Good Thinking: The Foundations of Probability and its Applications*. University of Minnesota Press. Unabridged republication by Dover 2009. [46, 202]

Gregory Grefenstette, 1994. *Explorations in Automatic Thesaurus Discovery*. Kluwer Academic Publishers. [12, 32, 37]

Ralph Grishman and John Sterling, 1993. Smoothing of automatically generated selectional constraints. In *Proceedings of the workshop on Human Language Technology*, pages 254–259. Association for Computational Linguistics. [43]

Zellig Harris, 1968. *Mathematical structures of language*. Wiley. [24]

Zellig S. Harris, 1954. Distributional Structure. *Word*. Reprinted 1964, 1970, 1985. [21, 25]

Donald Hindle, 1990. Noun classification from predicate-argument structures. In *Proceedings of the 28th Annual Meeting of the Association for Computational Linguistics*, pages 268–275. [12, 45]

E. T. Jaynes, 2003. *Probability Theory: The Logic of Science*. Cambridge University Press. [50]

Timo Järvinen and Pasi Tapanainen, 1997. A Dependency Parser for English. Technical Report TR-1, Department of General Linguistics, University of Helsinki. [54]

Timo Järvinen and Pasi Tapanainen, 1998. Towards an implementable dependency grammar. In Sylvain Kahane and Alain Polguère, editors, *Proceedings of the workshop "Processing of dependency-based grammars"*, pages 1–10. [54]

Lillian Lee, 1999. Measures of distributional similarity. In *Proceedings of 37th Annual Meeting of the ACL*, pages 25–32. [12, 34, 35, 36, 38, 42, 43, 46, 48, 94]

Lillian Lee, 2001. On the effectiveness of the skew divergence for statistical language analysis. In *Proceedings of AI and Statistics*. [35, 44, 46]

Lillian Lee and Fernando Pereira, 1999. Distributional Similarity Models: Clustering vs. nearest neighbors. In *Proceedings of 37th Annual Meeting of the ACL*, pages 33–40. [46]

Dekang Lin, 1998a. Automatic retrieval and clustering of similar words. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics (Coling-ACL '98)*. [29]

Dekang Lin, 1998b. An information-theoretic definition of similarity. In *Proceedings of International Conference on Machine Learning*. [12, 46]

Jianhua Lin, 1991. Divergence measures based on the Shannon entropy. *IEEE Transactions on Information Theory*, 37(1):145–151. [19, 42, 46, 47, 94, 170]

Krister Lindén, 2005. *Word sense discovery and disambiguation*. Ph.D. thesis, University of Helsinki. Publications of the department of general linguistics 37. [13, 204]

Krister Lindén and Jussi Piitulainen, 2004. Discovering synonyms and other related words. In *Proceedings of CompuTerm 2004, 3rd International Workshop on Computational Terminology*. Also included in (Lindén, 2005). [13]

Juha Makkonen and Jussi Piitulainen, 2001. Expanding document vectors in text categorization. In *Infotech Oulu International Workshop on Information Retrieval (IR2001)*, pages 52–60. [13]

Christopher D. Manning and Hinrich Schütze, 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press. [33, 38, 48, 59]

George A. Miller and Walter G. Charles, 1991. Contextual correlates of semantic similarity. *Language and Cognitive Processes*. [24, 25]

David S. Moore and George P. McCabe, 2003. *Introduction to the Practice of Statistics*. Freeman, fourth edition. [60]

Eugene A. Nida, 1975. *Componential Analysis of Meaning*. Mouton. [29]

F. R. Palmer, editor, 1968. *Selected Papers of J. R. Firth 1952–59*. Longmans. [202]

Philip Resnik, 1998. Semantic similarity in a taxonomy: an information-based measure and its application to problems of ambiguity in natural language. *Journal of Artificial Intelligence Research*. [27]

Herbert Rubenstein and John B. Goodenough, 1965. Contextual correlates of synonymy. *Communications of the ACM*. [24, 25]

Magnus Sahlgren, 2005. An introduction to random indexing. In *Proceedings of the Methods and Applications of Semantic Indexing Workshop at the 7th International Conference on Terminology and Knowledge Engineering, TKE 2005*. Available as PDF from author's homepage. [40]

Magnus Sahlgren, 2006. *The Word-Space Model: Using distributional analysis to represent syntagmatic and paradigmatic relations between words in high-dimensional vector spaces*. Ph.D. thesis, Stockholm University. [24]

Geoffrey Sampson, 2001. *Good–Turing frequency estimation without tears*, chapter 7, pages 94–121. Continuum. [59]

Robin Sibson, 1969. Information radius. *Probability Theory and Related Fields*, 14(2):149–160. In 1969, the journal was still Zeitschrift für Wahrscheinlichkeitstheorie und verwandte Gebiete, but now the paper is available on-line under the English name. [13, 19, 20, 46, 48, 94, 169]

John Sinclair, 1996. The Empty Lexicon. *International Journal of Corpus Linguistics*, 1(1):99–119. [31]

Pasi Tapanainen and Timo Järvinen, 1997. A non-projective dependency parser. In *Proceedings of the 5th Conference on Applied Natural Language Processing*, pages 64–71. [54]

Terry M. Therneau and Elizabeth J. Atkinson, 1997. An introduction to recursive partitioning using the RPART routines. [139, 148]

Julie Elizabeth Weeds, 2003. *Measures and Applications of Lexical Distributional Similarity.* Ph.D. thesis, University of Sussex. [12, 24]

Tsachy Weissman, Erik Ordentlich, Gadiel Seroussi, Sergio Verdu, and Marcelo J. Weinberger, 2003. Inequalities for the L_1 Deviation of the Empirical Distribution. Technical Report HLP-2003-97 (R.1), Information Theory Research Group, HP Laboratories. [42]